Predictive Analytics

Week 9: Classification I

Semester 2, 2018

Discipline of Business Analytics, The University of Sydney Business School

QBUS2820 content structure

1. Statistical and Machine Learning foundations and applications.

2. Advanced regression methods.
3. **Classification methods**.

4. Time series forecasting.

Week 9: Classification I

1. Classification
2. Introduction to decision theory for classification
3. K-nearest neighbours classifier
4. Logistic regression
5. Model evaluation for binary classification
6. Decision theory for binary classification (optional)

Readings: Chapters 2.2.3, 4.1, 4.2 and 4.3 of ISL.

Exercise questions: Chapter 4.7 of ISL, Q1, Q4, Q6, Q8 and Q9.

Classification

Classification

Consider the following business decision making scenarios.

1. Should we invest resources in acquiring and retaining a customer?

2. Should we offer a mortgage to a credit applicant?

3. Should we invest more resources to train an employee?

4. Should we place a bid to a sponsor an online search?

5. Should we investigate a transaction for possible fraud?

Classification

All these scenarios involve a **classification task**.

1. Do we predict that the customer will be profitable?

2. Do we predict that the applicant will repay the mortgage in full?

3. Do we predict that the employee will stay in the company?

4. Do we predict that the user will click on the ad and make a purchase?

5. Do we flag the transaction?

Classification

In classification, the response variable Y is **qualitative** or **categorical** that takes values in a finite unordered set $\mathcal{Y} = \{1, \ldots, C\}$, where C is the number of classes. Our task is to predict which class a subject belongs to based on input variables.

A **classifier** $\widehat{Y}(X)$ is a mapping from the input vector $\boldsymbol{x}$ to $\{1, \ldots, C\}$. A classifier is a prediction rule that assigns the subject to one of the classes, given the observed values of the predictors.

In the fraud detection example, our response variable may be flag $= \{\text{fraud}, \text{legitimate}\}$. We can code this variable as

$$Y = \begin{cases} 1 & \text{if fraud,} \\ 0 & \text{if legitimate.} \end{cases}$$

Introduction to decision theory for classification

Loss functions for classification

In classification, we represent the loss function by a $C \times C$ loss matrix $\mathbf{L}$. Each element of the loss matrix $L_{k\ell} = L(k, \ell)$ specifies the loss of classifying in class ℓ when the actual class is k.

In this section, we focus on a simple framework by considering the **zero-one loss function**.

Zero-one loss function (key concept)

The zero-one loss function is

$$L(y, \widehat{y}) = \begin{cases} 1 & \text{if } y \neq \widehat{y} \\ 0 & \text{if } y = \widehat{y}, \end{cases}$$

such that the loss is zero for a correct classification and one for a misclassification.

Classification risk

As before, our objective is to minimise the risk or expected loss of the classifier,

$$R(\widehat{Y}(X)) = E\left[L(Y, \widehat{Y}(X))\right].$$

You can think of the risk as the average loss across all subjects in the population (each subject has a pair of Y and X values).

By conditioning on the predictors, we can rewrite the risk as

$$R(\widehat{Y}(X)) = E_X\left(\sum_{c=1}^{C} E\left[L(c, \widehat{Y}(X))\right] P(Y = c|X)\right)$$

Classification

Classification models lead to estimated conditional probabilities $\widehat{P}(Y = c|X = \boldsymbol{x})$ for $c = 1, \ldots, C$. We then classify a test case to the class with highest estimated probability.

For binary classification, we therefore classify a subject as $\widehat{y} = 1$ if $\widehat{P}(Y = 1|X = \boldsymbol{x}) \geq 0.5$.

We compute the **misclassification** or **error rate** for the test data of size n as

$$\overline{\mathsf{Err}}_{\text{test}} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \widehat{y}_i).$$

K-nearest neighbours classifier

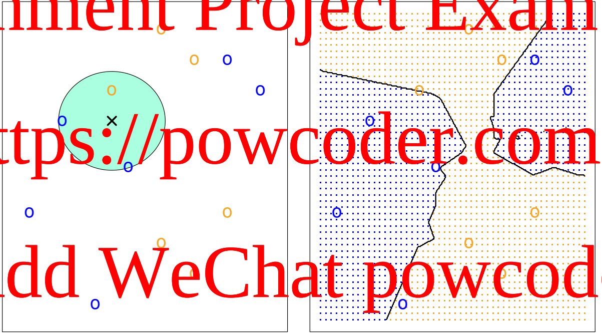
K-nearest neighbours classifier

The **K-nearest neighbours classifier** estimates the conditional probability for class c as

$$\widehat{P}(Y = c|X = \boldsymbol{x}) = \frac{1}{K} \sum_{\boldsymbol{x}_i \in \mathcal{N}_K} I(y_i = c)$$

for a training sample $\mathcal{D} = \{(y_i, \boldsymbol{x}_i)\}_{i=1}^{N}$

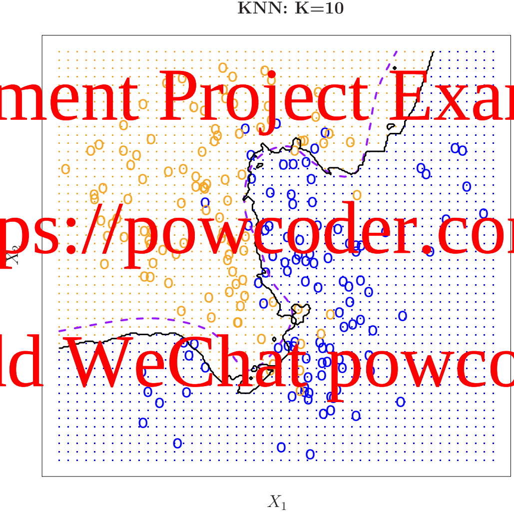
K-nearest neighbours classifier



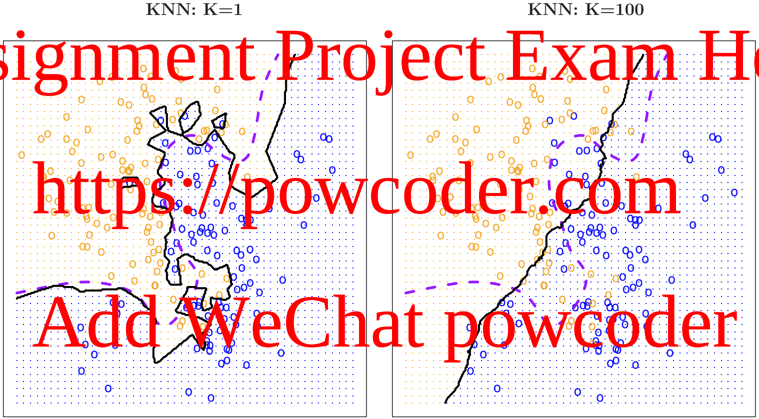
K-nearest neighbours classifier

- In words, the KNN method finds the K training input points which are closest to x, and computes the conditional probability as the fraction of those points that belongs to class c.
- Similarly to the KNN regression method, the KNN classifier is a direct nonparametric approximation to the Bayes classifier.
- The lower the K, the more flexible the decision boundary.
- As always, choosing the optimal level of flexibility is crucial. We use cross validation to select K.

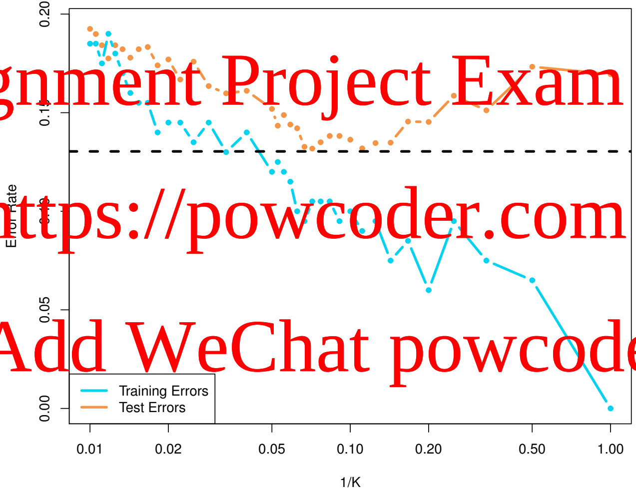
KNN classifier decision boundary



KNN classifier decision boundary



K-nearest neighbours classifier



Logistic regression

Regression models for classification

Suppose that we want the specify a discriminative model for binary classification. The response Y follows the Bernoulli distribution,

$$Y = \begin{cases} 1 \text{ with probability } P(Y = 1|X = \boldsymbol{x}) \\ 0 \text{ with probability } 1 - P(Y = 1|X = \boldsymbol{x}) \end{cases}$$

How to model the conditional probability $P(Y = 1|X = \boldsymbol{x})$ as a function of the predictors?

Regression models for classification

Since $P(Y=1|X=x) = E(Y|X=x)$, one option is to specify a linear regression model

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \varepsilon.$$

This is called the **linear probability model**. However, there are a few reasons why we want to move beyond this framework.

Why not the linear probability model?

1. There is no guarantee that a linear probability model will generate probabilities between zero and one, since the regression function $\beta_0 + \sum_{j=1}^{p} \beta_j x_j$ is unconstrained. In other words, the linearity assumption does not hold.

2. The Bernoulli distribution has variance $p(x)(1 - p(x))$. Hence, the linear probability model violates the classical assumption of constant error variance.

3. The linear probability approach is not robust to outliers.

4. The linear probability approach does not generalise to categorical responses with more than two classes.

Logistic regression (key concept)

The **logistic regression model** is

$$Y|X = \boldsymbol{x} \sim \text{Ber}(p(\boldsymbol{x})),$$

where

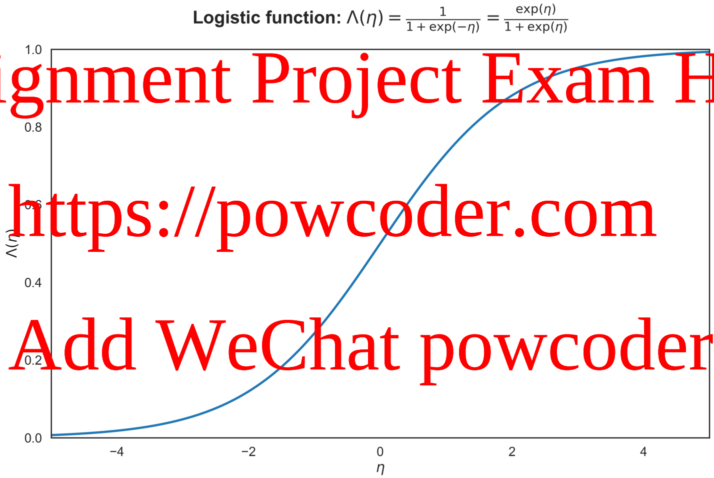
$$p(\boldsymbol{x}) = \frac{\exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_j)}{1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j x_j)}.$$

The **logistic function**

$$\frac{\exp(a)}{1 + \exp(a)} = \frac{1}{1 + \exp(-a)}$$

constrains the probability to be between zero and one.

Logistic function



Logistic Regression

Define the **odds ratio** as

$$\frac{p(\boldsymbol{x})}{1 - p(\boldsymbol{x})}.$$

We can show that

$$\frac{p(\boldsymbol{x})}{1 - p(\boldsymbol{x})} = \exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right).$$

The logistic regression model therefore specifies a linear model for the log odds,

$$\log\left(\frac{p(\boldsymbol{x})}{1 - p(\boldsymbol{x})}\right) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j,$$

where we call the left-hand side the logit transformation of the probability.

Prediction with logistic regression

Suppose that we a test point $\boldsymbol{x}_i$, if $(\widehat{\beta}_0 + \sum_{j=1}^{p} \widehat{\beta}_j x_{ij}) > 0$, then

$$\hat{P}(Y = 1 | X = \boldsymbol{x}_i) = \widehat{p}(\boldsymbol{x}_i) = \frac{\exp(\widehat{\beta}_0 + \sum_{j=1}^{p} \widehat{\beta}_j x_{ij})}{1 + \exp(\widehat{\beta}_0 + \sum_{j=1}^{p} \widehat{\beta}_j x_{ij})} > 0.5,$$

then prediction $\hat{y}_i = 1$. When to predict $\hat{y}_i = 0$?

If $(\widehat{\beta}_0 + \sum_{j=1}^{p} \widehat{\beta}_j x_{ij}) = 0$, we have the decision boundary of logistic regression.

Maximum likelihood estimation

We estimate the logistic regression model by maximum likelihood.

Recall that a Bernoulli random variable Y has probability mass function

$$p(y; \pi) = \pi^y (1 - \pi)^{1-y}.$$

In the context of the logistic regression model, the probability mass function for a training case i is therefore

$$p(y_i | x_i) = p(\boldsymbol{x}_i)^{y_i} (1 - p(\boldsymbol{x}_i))^{1-y_i}.$$

Maximum likelihood estimation

The likelihood function is

$$\begin{aligned}\ell(\boldsymbol{\beta}) &= p(y_1|\boldsymbol{x}_1)\, p(y_2|\boldsymbol{x}_2)\, \ldots\, p(y_N|\boldsymbol{x}_N) \\ &= \prod_{i=1}^{N} p(y_i|\boldsymbol{x}_i) \\ &= \prod_{i=1}^{N} p(\boldsymbol{x}_i)^{y_i}(1-p(\boldsymbol{x}_i))^{1-y_i}\end{aligned}$$

Maximum likelihood estimation

The log-likelihood is

$$L(\beta) = \log\left(\prod_{i=1}^{N} p(\boldsymbol{x}_i)^{y_i}(1-p(\boldsymbol{x}_i))^{1-y_i}\right)$$

$$= \sum_{i=1}^{N}\left(y_i \log(p(\boldsymbol{x}_i)) + (1-y_i)\log(1-p(\boldsymbol{x}_i))\right),$$

where $p(\boldsymbol{x}_i)$ can be calculated as before using logistic function.

The negative log-likelihood $-L(\beta)$ is known as the **cross-entropy loss function** or log loss in machine learning. Think the intuition of the above formula.

Maximum likelihood estimation

The MLE for the logistic regression model is

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}}\ L(\boldsymbol{\beta}),$$

where

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left(y_i \log(p(\boldsymbol{x}_i)) + (1 - y_i) \log(1 - p(\boldsymbol{x}_i)) \right)$$

Similar to Lecture 5, then it can be shown that:

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \boldsymbol{X}^T \left(\boldsymbol{y} - p(\boldsymbol{X}) \right)$$

Gradient Ascend in matrix form

Here we have:

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$$p(\boldsymbol{X}) = \begin{bmatrix} p(\boldsymbol{x}_1) \\ p(\boldsymbol{x}_2) \\ \vdots \\ p(\boldsymbol{x}_N) \end{bmatrix}$$

Hence gradient ascent in matrix form for logistic regression is:

$$\boldsymbol{\beta} := \boldsymbol{\beta} + \alpha \boldsymbol{X}^T(\boldsymbol{y} - p(\boldsymbol{X}))$$

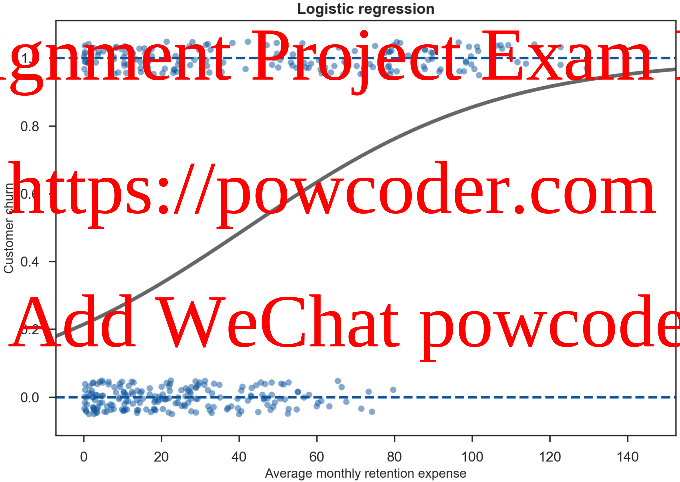
- Note the size of each matrix and vector in the above formula.
- Try to implement this in Python.

Maximum likelihood estimation

Optimisation. Setting the partial derivatives of the log likelihood to zero leads to estimation equations that are nonlinear in the coefficients. We therefore use numerical optimisation routines to obtain the estimates.

Statistical inference. We can conduct statistical inference for the logistic regression model using the large sample theory for ML estimation or the Bootstrap method.

Example: customer churn data



Customer churn data

Response: whether the customer had churned by the end of the observation period.

Predictors

1. Average number of dollars spent on marketing efforts to try and retain the customer per month.
2. Total number of categories the customer has purchased from.
3. Number of purchase occasions.
4. Industry: 1 if the prospect is in the B2B industry, 0 otherwise.
5. Revenue: annual revenue of the prospect's firm.
6. Employees: number of employees in the prospect's firm.

Observations: 500.

Source: Kumar and Petersen (2012).

Example: customer churn data


```
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                  350
Model:                          Logit   Df Residuals:                      348
Method:                           MLE   Df Model:                            1
Date:                                   Pseudo R-squ.:                  0.1319
Time:                                   Log-Likelihood:                -208.99
converged:                       True   LL-Null:                       -240.75
                                        LLR p-value:                 1.599e-15
===============================================================================
                  coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept      -1.2959      0.189     -6.866      0.000      -1.666      -0.926
Avg_Ret_Exp     0.0308      0.004      7.079      0.000       0.022       0.039
===============================================================================
```

Example: customer churn

No we can predict the probability that a customer with average retention expenses of 100 will churn.

$$
\begin{aligned}
\widehat{p} &= \frac{\exp(\widehat{\beta}_0 + \widehat{\beta}_1 \times 100)}{1 + \exp(\widehat{\beta}_0 + \widehat{\beta}_1 \times 100)} \\
&= \frac{\exp(-1.296 + 0.031 \times 100)}{1 + \exp(-1.296 + 0.031 \times 100)} \\
&= 0.856
\end{aligned}
$$

Decision boundary? If we have two predictors x_1 and x_2, how the decision boundary will be looked like?

Regularised logistic regression

Regularised risk minimisation applies to logistic regression. With an ℓ_1 penalty as in the lasso, we solve the minimisation problem

$$\min_{\boldsymbol{\beta}} \; -L(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j|,$$

where

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left(y_i \log(p(\boldsymbol{x}_i)) + (1 - y_i) \log(1 - p(\boldsymbol{x}_i)) \right)$$

Subset selection and dimension reduction with principal components also extend to logistic regression in a straightforward way.

Model evaluation for binary classification

Decision rule (key concept)

More generally, the decision to classify a subject as positive or negative is based on a decision rule

$$\delta(\boldsymbol{x}) = \begin{cases} 1 & \text{if } P(Y = 1|X = \boldsymbol{x}) > \tau. \\ 0 & \text{if } P(Y = 1|X = \boldsymbol{x}) \leq \tau. \end{cases}$$

where τ is a decision threshold parameter.

Confusion matrix (key concept)

A **confusion matrix** counts the number of true negatives, false positives, false negatives, and true positives for the test data.

		Classification (Prediction)		
		$\widehat{Y}=0$	$\widehat{Y}=1$	Total
Actual	$Y=0$	True negatives (TN)	False positives (FP)	N
	$Y=1$	False negatives (FN)	True positives (TP)	P
	Total	Negative predictions	Positive predictions	

Estimating the generalisation error

- Estimating the generalisation error is straightforward using the loss and confusion matrices.

- As always, it is important to quantify the uncertainty in the estimate by reporting the standard error or doing interval estimation.

- For the rest of this section, we discuss important concepts for assessing binary classification models.

Sensitivity and specificity (key concepts)

The **sensitivity**, recall or true positive rate is

$$P(\widehat{Y}=1|Y=1)=\frac{\mathsf{TP}}{\mathsf{TP}+\mathsf{FN}}=\frac{\text{True positives}}{\text{Actual positives}}.$$

The **specificity** is

$$P(\widehat{Y}=0|Y=0)=\frac{\mathsf{TN}}{\mathsf{TN}+\mathsf{FP}}=\frac{\text{True negatives}}{\text{Actual negatives}}.$$

False positive and false negative rates

The **false positive rate** (FPR) is

$$P(\widehat{Y}=1|Y=0)=\frac{\mathsf{FP}}{\mathsf{TN}+\mathsf{FP}}=\frac{\text{False positives}}{\text{Actual negatives}}=1-\text{Specificity}.$$

The **false negative rate** (FNR) is

$$P(\widehat{Y}=0|Y=1)=\frac{\mathsf{FN}}{\mathsf{TP}+\mathsf{FN}}=\frac{\text{False negatives}}{\text{Actual positives}}=1-\text{Sensitivity}.$$

Trade-off between sensitivity and specificity (key concept)

- There is a trade-off between sensitivity and specificity, since a classifier can always obtain maximum sensitivity (specificity) by setting $\gamma = 0$ ($\gamma = 1$) and automatically returning positive (negative).

- Equivalently, this is a trade-off between sensitivity and achieving a lower false positive rate.

Example: credit scoring

- Decreasing the threshold makes the loan decisions more lenient, leading to loans to customers with lower probability of full repayment.

- Issuing additional loans will increase both the number of true positives (higher sensitivity) and false positives/defaults (lower specificity).

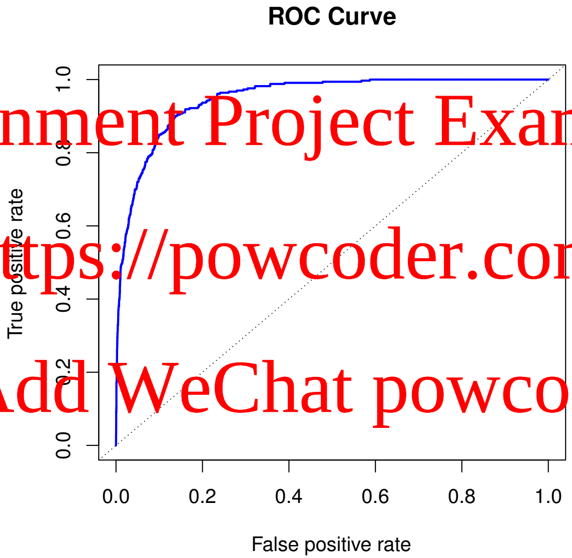
ROC curve (key concept)

A **receiver operating characteristic** or **ROC** curve plots the sensitivity against specificity or the false positive rate for a range of threshold values τ.

We can read the the ROC plot as telling us the false positive rate that we need to accept to obtain a given level of sensitivity.

We often summarise the quality of ROC curve as a single number using the **area under the curve** or **AUC**. Higher AUC scores are better, with a maximum of one.

ROC curve



ROC Curve
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

Imbalanced classes

Many classification scenarios (such as fraud detection) concern rare events, leading to a very large proportion of negatives in the data.

In this situation we say that the classes are highly **imbalanced**.

The specificity is not very informative for these problems, as it will tend to be high regardless of the quality of the classifier (nearly all transactions are legitimate and classified as such).

Precision (key concept)

In the imbalanced scenario, we are usually more interested in the proportion of detections that are actually positive. We define the **precision** as

$$P(Y = 1|\widehat{Y} = 1) = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{True positives}}{\text{Positive classifications}}$$

The **false discovery rate** (FDR) is one minus the precision.

$$P(Y = 0|\widehat{Y} = 1) = \frac{\text{FP}}{\text{TP} + \text{FP}} = \frac{\text{False positives}}{\text{Positive classifications}}$$

Precision recall curve

A **precision recall curve** plots the precision against the recall (sensitivity) as we vary the threshold T. The mean precision (averaging over recall values) approximates the area under the precision recall curve.

Example: transaction fraud detection

- In fraud detection, the bank is concerned that too many false alarms (low precision) would lead to high costs of investigating flagged transactions.

- The bank will weigh this cost against the savings from catching fraudulent transactions.

- Therefore, the financial institution is primarily interested in the precision recall curve.
- Increasing τ reduces the number of false alarms.

Review questions

- What is classification?
- What is a zero-one loss?
- What is the misclassification rate?
- Explain the KNN classifier.
- How do we formulate a decision rule for binary classification?
- What is a confusion matrix? Write down how the matrix looks like.
- What are sensitivity, specificity, and precision?
- Why is there a trade-off between sensitivity and specificity?

Decision theory for binary classification (optional)

Classification outcomes

In most business problems, there are distinct losses associated with each classification outcome. Consider for example the case of transaction fraud detection.

		Classification	
		Legitimate	**Fraud**
Actual	**Legitimate**	No loss	Investigation cost
	Fraud	Fraud loss	Fraud loss avoided

The cost of investigating a suspicious transaction is likely to much lower than the loss in case of fraud.

Classification outcomes

We use the following terminology.

		Classification	
		$\widehat{Y} = 0$	$\widehat{Y} = 1$
Actual	$Y = 0$	True negative	False positive
	$Y = 1$	False negative	True positive

Loss matrix (key concept)

The context will specify a **loss matrix** or **cost-benefit matrix** for classification as follows.

		Classification	
		$\widehat{Y}=0$	$\widehat{Y}=1$
Actual	$Y=0$	L_{TN}	L_{FP}
	$Y=1$	L_{FN}	L_{TP}

Example: credit scoring

In credit scoring, we want to classify a loan applicant as creditworthy ($Y = 1$) or not ($Y = 0$) based on the probability that the customer will not default.

		Classification	
		$\widehat{Y} = 0$	$\widehat{Y} = 1$
Actual	$Y = 0$	Default loss avoided	Default loss
	$Y = 1$	Profit opportunity lost	Profit

A false positive is a more costly error than a false negative for this business scenario. Our decision making should therefore take this into account.

Optimal decision (key concept)

Our decision problem is therefore to select an optimal threshold τ.

$$\tau^* = \underset{0<\tau<1}{\text{argmin}}\; E\left[L(Y, \delta_\tau(\boldsymbol{x}))|X = \boldsymbol{x}\right].$$

Let $\pi = P(Y = 1|X = \boldsymbol{x})$ to simplify the notation. We compare the expected loss from each decision,

$$E\left[L(Y, \delta_\tau(\boldsymbol{x}))|X = \boldsymbol{x}\right] = \begin{cases} \pi L_{\mathsf{TP}} + (1 - \pi)L_{\mathsf{FP}} \text{ if } \delta_\tau(\boldsymbol{x}) = 1, \\ \pi L_{\mathsf{FN}} + (1 - \pi)L_{\mathsf{TN}} \text{ if } \delta_\tau(\boldsymbol{x}) = 0. \end{cases}$$

Optimal decision (key concept)

The optimal decision threshold corresponds to the probability value π such that the loss from a positive or negative classification is equal:

$$\tau^* L_{\mathsf{TP}} + (1-\tau^*) L_{\mathsf{FP}} = \tau^* L_{\mathsf{FN}} + (1-\tau^*) L_{\mathsf{TN}}$$

Therefore, the optimal threshold is

$$\tau^* = \frac{L_{\mathsf{FP}} - L_{\mathsf{TN}}}{L_{\mathsf{FP}} + L_{\mathsf{FN}} - L_{\mathsf{TP}} - L_{\mathsf{TN}}}$$

Example: zero-one loss

With the zero-one loss, we have that $L_{\mathsf{FP}} = L_{\mathsf{FN}} = 1$ and $L_{\mathsf{TP}} = L_{\mathsf{TN}} = 0$. Therefore,

$$\tau^* = \frac{L_{\mathsf{FP}} - L_{\mathsf{TN}}}{L_{\mathsf{FP}} + L_{\mathsf{FN}} - L_{\mathsf{TP}} - L_{\mathsf{TN}}} = \frac{1}{2}$$

Example: credit scoring

In the credit scoring example, we have that $L_{\mathsf{TP}} = -L_{\mathsf{FN}}$ (profit equals missed profit) and $L_{\mathsf{TN}} = -L_{\mathsf{FP}}$ (avoided default loss equals default loss).

Therefore,

$$\tau^* = \frac{L_{\mathsf{FP}} - L_{\mathsf{TN}}}{L_{\mathsf{FP}} + L_{\mathsf{FN}} - L_{\mathsf{TP}} - L_{\mathsf{TN}}} = \frac{L_{\mathsf{FP}}}{L_{\mathsf{FN}} + L_{\mathsf{FP}}}.$$

Example: credit scoring

Optimal threshold for loan decision:

$$\tau^* = \frac{L_{\mathsf{FP}}}{L_{\mathsf{FN}} + L_{\mathsf{FP}}}.$$

We expect that the loss from default to be much higher than the profit from a loan to a creditworthy customer ($L_{\mathsf{FP}} >> L_{\mathsf{FN}}$), leading to a high threshold τ^*.

It is only worth it to lend to customers that have a high probability of repayment.