Machine Learning 2023

Final Exam

WITH ANSWERS

27 March 2023, 12:15–14:30

The questions start on the next page. **Please do not open the exam booklet until the scheduled starting time.**

Feel free to write on the exam booklet, and take it with you afterwards. The exam will be made available on Canvas later today, together with the correct answers. You can copy your final answers onto the booklet, so you can check how you did.

There are 40 questions, worth 1 point each.

To pass the examination, your total points for the exam plus those for the quizzes should be around 50–55. The exact pass mark will be decided *after* the exam results have been analysed.

Rules:

- You are allowed to use a calculator or graphical calculator.
- You are *not* allowed to use your phone or smartphone.
- The exam is closed-book.
- You are allowed to use the formula sheet provided through Canvas.
- The formula sheet should not have any writing on it, except in the cheat sheet box. Anything inside the box should be written by hand.

Recall questions

1. How are *hyperparameters* defined?

   A They are parameters of a learning algorithm that are set before we start training on the data. ✓
   B They are parameters of which the values are found by the training algorithm itself.
   C They are parameters that make the models train faster.
   D They are parameters of which the values are set after training the model.

   Explained in many places, for example here

2. In principal component analysis, which is **true**?

   A The first principal component is the direction in which the variance is the smallest.
   B The first principal component is the direction that maximizes the reconstruction loss.
   C The first principal component is the direction that minimizes the reconstruction loss. ✓
   D The first principal component is the direction in which the bias is the smallest.

   The first PC minimizes the reconstruction loss (this is used in our main derivation starting here). It also maximizes the variance, as explained starting here.

3. What is a valid reason to prefer *gradient descent* over *random search*?

   A My model is easily differentiable.✓
   B I need to be sure that I find the global minimum.
   C My loss function is not smooth.
   D There is some computation between the output of my model and my loss function, which I do not control.

   Gradient descent is defined starting here. It requires the objective function (the loss) to be differentiable. It is not guaranteed to find the global optimum. A non-smooth loss surface is problematic for gradient descent. See the discussion here. If there are parts of the computation graph that we do not control, as in Reinforcement Learning, then that implies that our model is not differentiable, and we cannot apply gradient descent without special tricks. See for instance this slide.

4. If we have a feature that is categorical, but our model requires numeric features, we can turn the categoric feature into one or more numeric features either by **integer coding**, or by **one-hot coding**. Which is **true**?

A The benefit of integer coding is that it assigns each value a separate feature.

B Integer coding results in more features than one-hot coding.

C The feature(s) introduced by integer coding are are valued between 0 and 1.

D The feature(s) introduced by one-hot coding are valued 0 or 1. ✓

Explained here `https://mlvu.github.io/lecture05/#slide-041`

5. Two classes that are not linearly separable may *become* linearly separable if we add a new feature that is derived from one or more of the other features. Why?

   A This approach removes outliers.
   B A linear decision boundary in the new space may represent a nonlinear boundary in the old one.✓
   C Adding a new feature derived from the existing ones normalizes the data.
   D This approach makes a non-differentiable loss function differentiable.

   Some examples are given here. This does not remove any data, so no outliers are removed. It also does not necessarily lead to data with a specific scale (normalization). It does not affect the loss function at all (only the features).

6. In the Expectation-Maximization algorithm for Gaussian mixture models, we train a model consisting of several *components*, and we compute various *responsibilities*. Which is **true**?

   A Each component is a normal distribution. Its responsibility for a point is the normalized probability density it assigns the point. ✓
   B Each component is a normal distribution. Its responsibility is the integer number of points it generates.
   C Each component is one of the features of the data. Its responsibility for a point is the normalized probability density it assigns the point.
   D Each component is one of the features of the data. Its responsibility is the integer number of points it generates.

   Components are explained here here. The responsibilities are explained here.

7. Lise is given an imbalanced dataset for a binary classification task. Which one of the following options is useful solution to this problem?

   A She can oversample her majority class by sampling with replacement.
   B She can use accuracy to measure the performance of her classifier.
   C She can use SMOTE to augment her training data. ✓
   D She can augment the feature set by adding the multiplication of some features.

   Oversampling the majority class would make the problem worse. Accuracy is a bad measure to use in the face of class imbalance. Augmentation of the features will change nothing about the class balance. SMOTE is discussed here here.

8. What is **true** about *logistic regression*?

A Logistic regression is a linear, discriminative classifier with a cross-entropy loss function.✓
B Logistic regression finds the maximum margin hyperplane when applied to data with well-separated classes.
C Logistic regression is identical to the least-square linear regression.
D Thanks to the log-loss function, logistic regression can learn nonlinear decision boundaries.

Logistic regression is linear: we apply a nonlinear sigmoid function, but the decision boundary remains linear. It is also disciminative, and a classifier (despite the name).

It does not use the maximum-margin hyperplane criterion (that's the SVM), and it is substantially different from the least squares linear regression (it's a classifier, for starters).

9. Neural networks usually contain *activation functions*. What is their purpose?

   A They are used to compute a local approximation of the gradient.
   B They are applied after a linear transformation, so that the network can learn nonlinear functions. ✓
   C They control the magnitude of the the step taken during an iteration of gradient descent.
   D They function as a regularizer, to combat overfitting.

   Activation functions are explained here. The slides previous to that explain that a network without them (containing only linear functions) cannot learn anything more than a linear function.

10. I am training a generator network to generate faces. I take a random sample, compare it to a randomly chosen image form the data, and backpropagate the error.

    When training is finished, all samples from the network look like the average over all faces in the dataset.

    What name do we have for this phenomenon?

    A Multiple testing
    B Overfitting
    C Dropout
    D Mode collapse ✓

    Mode collapse (in the context of generative modelling) is explained here. Multiple testing is a problem in statistics. It happens in ML when we re-use our test set. Overfitting is a general problem of memorization in all ML models, and dropout is a trick you can apply to neural networks to stop overfitting.

11. Which is **true**?

    A A maximum likelihood objective for least-squares regression does not provide a smooth loss surface.
    B The least-squares loss for linear regression can be derived from a maximum likelihood objective. ✓
    C Linear regression can be performed with a maximum likelihood objective but the results will be different from the least-squares loss.
    D The loss function used in logistic regression is derived from assuming a normal distribution on the residuals.

    This is explained here. Maximum likelihood is often used to provide smooth losses (such as the log-loss, which is used as a proxy for the accuracy). Answer C is the opposite of B and the logistic regression loss is derived from assuming a Bernoulli distribution on the outputs, not a Normal distribution.

12. What is an important difference between regular recurrent neural networks (RNNs) and LSTM neural networks?

    A All LSTMs have a *forget gate*, allowing them to ignore parts of the cell state right away. ✓

    B All RNNs have a *forget gate*, allowing them to ignore parts of of the cell state right away.

    C LSTMs have a vanishing gradient problem, RNNs don't.

    D RNNs can be turned into variational autoencoders, LSTMs can't.

    See this slide. The vanishing gradient problem applies to RNNs and was *solved* by LSTMs. Both RNNs and LSTMs can be used in a VAE, since the VAE is defined independent of the specific architecture used for the encoder and decoder.

13. Which of the following is an ensemble method?

A Random forest
B Gradient boosting
C AdaBoost
D All of the above ✓

See `https://mlvu.github.io/lecture10/#video-037`

14. William has a dataset of many recipes and many ingredients. He doesn't know anything about the recipes except which ingredients occur in each, and he doesn't know anything about the ingredients except in which recipes they occur.

He'd like to predict for a pair of an ingredient and a recipe (both already in the data) whether the recipe would likely be improved by adding that ingredient.

Which is **true**?

A He could model the recipes as instances with their ingredients as a single categorical feature, and solve the problem with a decision tree.

B He could model the ingredients as instances and their recipes as a single categorical feature, and solve the problem with a decision tree.

C He could model this as a matrix decomposition problem. ✓

D This problem requires a sequence-to-sequence model.

This is one of the examples given in this slide. Matrix decomposition is another word for the embedding method discussed in this lecture.

Combination questions

15. The mean squared error (MSE) loss function $\sum_i (y_i - t_i)^2$ and the mean absolute error (MAE) loss function $\sum_i |y_i - t_i|$ are two popular loss functions. *Here the sum is over n instances, y_i is the model output for instance i and t_i is the training label.*

Which would be a a reason for preferring the MAE over the MSE?

A In MAE, the negative and positive differences do not cancel out in the sum.

B The mean of the error $y_i - t_i$ minimizes the MAE.

C The MAE is less sensitive to outliers than the MSE. ✓

D There is no advantage in using the MAE over the MSE.

The fact that the MAE is less sensitive to outliers is explained here. The use of MAE as a loss function is referenced in various places. For instance here and here.

16. We are choosing a new basis for our data. We decide to use an *orthonormal* basis. What is the advantage of having an orthonormal

basis?

A It ensures that the basis vectors are equal to the principal components.
B It ensures that the basis vectors are orthogonal to the principal components.
C It ensures that the inverse of the basis matrix is equal to its transpose.✓
D It ensures that the data is automatically whitened in the new basis.

Orthonormal bases are explained here.

17. Which of these statements about principal component analysis is **false**?

A The first principal component provides the direction of greatest variance.
B It is a supervised method. ✓
C It can be used for dimensionality reduction.
D It can be used for data pre-processing.

PCA is introduced as a dimensionality reduction method. Dimensionality reduction has many uses, and pre-processing is one of them. The first PC is defined as the direction of greatest variance (see also question 2). This leaves answer B: we can apply PCA to a dataset without knowing the labels, so it is an *un*supervised method.

18. We want to represent color videos in a deep learning system. Each is a series of frames, with each frame an RGB image. Which is the most natural representation for *one* such video?

A As a 1-tensor.
B As a 2-tensor.
C As a 3-tensor.
D As a 4-tensor. ✓

As explained here, RGB images can be stored in a 3-tensor. For a sequence of such images, we'd need one additional dimension, giving us a 4-tensor.

19. What is **not true** about the naive Bayes classifier?

A The naive Bayes classifier can be applied to multi-class classification.
B The naive Bayes classifier assumes that the features are independent. ✓
C To avoid zero probabilities in the naive Bayes classifier, one can use the Laplace smoothing technique.
D The naive Bayes classifier is a probabilistic generative classifier.

Naive Bayes does not assume that the features are independent. It assumes that there a *conditionally* independent (when conditioned on the class). All other answers are true.

20. One way to think of a convolution layer is as a fully connected layer, with some extra constraints. Which is **not** one of these constraints?

A Some of the weights are forced to have the same value.
B Some of the connections are removed.
C The L2 norm of the weights is limited to a maximum value. ✓
D All of the above are part of a convolution layer.

We introduce the convolutional layer by starting with a fully connected one and transforming it to reduce the number of weights. We remove many weight, and force others to have the same value. We do not place any restrictions on their norm.

21. Which answer contains methods that can *all* be used as sequence-to-sequence layers?

A Convolutions, RNN, LSTM✓
B Gradient boosting, LSTM, Deep Q-Learning
C Convolutions, Word2Vec, Gradient boosting
D RNN, Deep Q-Learning, Word2Vec

Seuqnece-to-sequence layers are explained here. RNNs, LSTMs and Convolutions can all be used as sequence-to-sequence layers. The other methods are not neural network layers. Word2Vec could *possibly* be interpreted as such, but it doesn't really work.

22. How are the Gaussian Mixture Model (GMM) and the Mixture Density Network (MDN) related?

    A The MDN is an alternative to the GMM that can also describe complex distributions, but that doesn't use Gaussians.
    B The GMM is an alternative to the MDN that can also describe complex distributions, but that isn't a mixture model.
    C The MDN is a neural network which uses the GMM as its output distribution. ✓
    D The GMM is a neural network which uses the MDN as its output distribution.

    The MDN is a neural network with a GMM as an output distribution. This means that both are a mixture model and both use Gaussians. The GMM is described starting here and the MDN is described starting here.

23. I have a dataset of politicians in the European parliament and which laws they voted for and against. The record is incomplete, but I have some votes for every law and for every politician.

    I would like to predict, for a new law, which politicians will vote for and which will vote against. I plan to model this as a recommender system using *matrix factorization*.

    Which is **true**?

    A This is not a good model, because there are too many classes and not enough instances.
    B This is not a good model, because there are not enough classes, and too many instances.
    C I would have to deal with the cold start problem, because for the new law I don't have any voting information. ✓
    D I would have to deal with the cold start problem, because the voting record is incomplete.

    Matrix factorization does not require class or instances. The fact that the voting record is not incomplete is not a problem. compare this to movie recommendation: the fact that we don't have annotations for all user/movie pairs doesn't pose a problem. The fact that the law is new *does* pose a problem. It means that we have no connections from the new law to any politicians. Just like when we add a new movie to our database, we don't have likes for it yet. This is the cold start problem.

24. Which is a good principle to decide what feature to use for the next split in a decision tree?

A After the split we want the probability of the majority class to be as low as possible.
B After the split we want the probability of the majority class to be as high as possible.
C After the split we want the entropy of the class distribution to be as low as possible.✓
D After the split we want the entropy of the class distribution to be as high as possible.

Explained here.

25. What is the difference between *inductive* and *transductive* learning?

A In transductive learning the model is allowed to see the labels of the test data during training.
B In inductive learning the model is allowed to see the labels of the test data during training.
C In transductive learning, the model is allowed to see the features of the test data during training. ✓
D In inductive learning, the model is allowed to see the features of the test data during training.

Defined here

Application questions

We have the following training set:

	x_1	x_2	label
a	1	3	Pos
b	2	2	Pos
c	3	1	Neg
d	6	4	Neg
e	5	5	Pos
f	4	6	Pos
g	7	7	Neg

For the following questions, it helps to draw the data and the classification boundary in feature space.

We use a linear classifier defined by

$$c(x_1, x_2) = \begin{cases} \text{Pos} & \text{if } -x_1 + 0 \cdot x_2 > -2 \\ \text{Neg} & \text{otherwise.} \end{cases}$$

26. If we turn c into a *ranking* classifier, how does it rank the points, from most Negative to most Positive?

A c b a d e f g
B g f e d a b c
C g d e f c b a ✓
D a b c f e d g

27. How many ranking errors does the classifier make?

A 1
B 2✓
C 3
D 4

If you answered 3, make sure you understand the difference between a *ranking error* and a *classification error*. Classification errors as instances that the classifier classifies incorrectly (in this case the instances b, e, and f) and ranking errors are *pairs* of positive and negative instances that a ranking classifier ranks the wrong way around. In this case (c, e) and (c,f)

28. If we draw a coverage matrix (as done in the slides), what proportion of the cells will be red?

    A $\frac{1}{2}$ **B** $\frac{1}{3}$ **C** $\frac{1}{4}$ **D** $\frac{1}{6}$✓

    The size of the coverage matrix is the number of positives times the number of negatives ($4 \times 3 = 12$). The number of red cells is the number of ranking errors (2) so the proportion of red cells is $2/12 = 1/6$

We will use the backpropagation algorithm to find the derivative of the function

$$f(x) = \frac{\sin\left(x^2\right)}{x^3}$$

with respect to x.

First, we break the function up into modules:

$$a = x^2$$
$$b = \sin a$$
$$c = x^3$$
$$f = \frac{b}{c}$$

29. What should be in the place of the dots?

    A $b = \sin x,\ c = x^2$

    B $b = \sin x,\ c = x^3$

    C $b = \sin a,\ c = x^2$

    D $b = \sin a,\ c = x^3$ ✓

30. In terms of the *local* derivatives. Which is the correct expression for the derivative $\frac{\partial f}{\partial x}$?

A $\frac{2x\cos a}{c} - \frac{3bx^2}{c^2}$✓

B $\frac{2x\cos a}{c} + \frac{3bx^2}{c^2}$

C $\frac{2x\cos a}{c} - \frac{3bx}{c^2}$

D $\frac{2x\cos a}{c} + \frac{3bx}{c^2}$

We have trained a linear support vector machine, and found the following weights and bias:

$$\boldsymbol{w} = \begin{pmatrix} 1.5 \\ -2 \end{pmatrix}, \quad b = -1. \tag{1}$$

31. Which of the following points are *support vectors* for the trained model?

A $y_1 = 1, \mathbf{x}_1 = (2, 1.5)$ $y_2 = 1, \mathbf{x}_2 = (0, 1)$

B $y_1 = 1, \mathbf{x}_1 = (1, -0.25)$ $y_2 = -1, \mathbf{x}_2 = (0, 1)$

C $y_1 = -1, \mathbf{x}_1 = (2, 1.5)$ $y_2 = 1, \mathbf{x}_2 = (0, -1)$✓

D $y_1 = -1, \mathbf{x}_1 = (2, 0.5)$ $y_2 = 1, \mathbf{x}_2 = (1, 0.75)$

32. Consider two points: $\mathbf{x}_1 = (1, 0)$ and $\mathbf{x}_2 = (-0.5, -1)$. How are these two points classified?

A Both positive.✓

B $\mathbf{x}_1$ is positive, $\mathbf{x}_2$ is negative.

C $\mathbf{x}_1$ is negative, $\mathbf{x}_2$ is positive.

D Both negative.

We are given a dataset of email, labeled spam and ham. The total number of words in the spam and ham datasets is is 50 000 and 200 000, respectively.

On this dataset, we want to train a *first-order Markov model* as a basis for a Bayes classifier to detect spam.

The following table shows the frequency of several bigrams and unigrams in the dataset.

uni- and bigrams	frequency	
	spam	ham
you	5 000	5 000
won	3 000	1 000
a	9 000	9 000
prize	2 000	1 000
you won	300	100
won a	300	100
a prize	300	100

We have two *priors* on how likely an email is to be spam. **Prior 1** says that an email is as likely to be spam as ham. **Prior 2** says that the probability that a given email is spam is 1%.
Consider the following email:

email: you won a prize

33. How is the email classified?
    A With both priors as ham.
    B With both priors as spam.
    C Under prior 1 as spam, under prior 2 as ham. ✓
    D Under prior 1 as ham, under prior 2 as spam.

$$
\begin{aligned}
\mathrm{p}(\mathrm{you}, \mathrm{won}, \mathrm{a}, \mathrm{prize} \mid \mathrm{spam}) = \mathrm{p}(\mathrm{prize} \mid \mathrm{a}, \mathrm{spam}) \cdot & \quad \frac{300}{9\,000} = \frac{1}{30} \\
\mathrm{p}(\mathrm{a} \mid \mathrm{won}, \mathrm{spam}) \cdot & \quad \frac{300}{3\,000} = \frac{1}{10} \\
\mathrm{p}(\mathrm{won} \mid \mathrm{you}, \mathrm{spam}) \cdot & \quad \frac{300}{5\,000} = \frac{3}{50} \\
\mathrm{p}(\mathrm{you} \mid \mathrm{spam}) & \quad \frac{5\,000}{50\,000} = \frac{1}{10}
\end{aligned}
$$

$$
= \frac{1}{30}\frac{1}{10}\frac{3}{50}\frac{1}{10} = \frac{3}{3 \cdot 5 \cdot 10^4} = \frac{1}{5} \cdot 10^{-4}
$$

$$
\begin{aligned}
\mathrm{p}(\mathrm{you}, \mathrm{won}, \mathrm{a}, \mathrm{prize} \mid \mathrm{ham}) = \mathrm{p}(\mathrm{prize} \mid \mathrm{a}, \mathrm{ham}) \cdot & \quad \frac{100}{9\,000} = \frac{1}{90} \\
\mathrm{p}(\mathrm{a} \mid \mathrm{won}, \mathrm{ham}) \cdot & \quad \frac{100}{1\,000} = \frac{1}{10} \\
\mathrm{p}(\mathrm{won} \mid \mathrm{you}, \mathrm{ham}) \cdot & \quad \frac{100}{5\,000} = \frac{1}{50} \\
\mathrm{p}(\mathrm{you} \mid \mathrm{ham}) & \quad \frac{5\,000}{200\,000} = \frac{1}{40}
\end{aligned}
$$

$$
= \frac{1}{90}\frac{1}{10}\frac{1}{50}\frac{1}{40} = \frac{1}{5 \cdot 4 \cdot 9 \cdot 10^4} = \frac{1}{180} \cdot 10^{-4}
$$

To get the classification we multiply both by their prior and check which is higher. Under prior 1, both prior probabilities are equal so the classification is spam.

Under prior 2, we get

$$p(\text{spam} \mid E) \propto \frac{1}{5} \cdot 10^{-4} \cdot \frac{1}{100} = \frac{1}{5} \cdot 10^{-6}$$

and

$$p(\text{ham} \mid E) \propto \frac{1}{180} \cdot 10^{-4} \cdot \frac{99}{100} = \frac{99}{180} \cdot 10^{-6} = \frac{11}{20} \cdot 10^{-6}$$

so the classification is ham.

34. Under prior 1, what is the probability that the email is spam?

A $\frac{1}{37}$ **B** $\frac{36}{37}$ ✓ **C** $\frac{1}{27}$ **D** $\frac{26}{27}$

$$\begin{aligned} p(\text{spam} \mid E) &= \frac{p(E \mid \text{spam})p(\text{spam})}{p(E \mid \text{spam})p(\text{spam}) + p(E \mid \text{ham})p(\text{ham})} \\ &= \frac{\frac{1}{5} \cdot 10^{-4} \cdot \frac{1}{2}}{\frac{1}{5} \cdot 10^{-4} \cdot \frac{1}{2} + \frac{1}{180} \cdot 10^{-4} \cdot \frac{1}{2}} \\ &= \frac{\frac{1}{5}}{\frac{1}{5} + \frac{1}{180}} = \frac{\frac{36}{180}}{\frac{36}{180} + \frac{1}{180}} = \frac{36}{37} \end{aligned}$$

35. Under prior 2, what is the *probability ratio* of ham over spam for the email?

A $\frac{4}{11}$ **B** $\frac{11}{4}$ ✓ **C** $\frac{1}{36}$ **D** 36

$$\begin{aligned} \frac{p(\text{ham} \mid E)}{p(\text{spam} \mid E)} &= \frac{p(E \mid \text{ham})p(\text{ham})\frac{1}{p(E)}}{p(E \mid \text{spam})p(\text{spam})\frac{1}{p(E)}} \\ &= \frac{\frac{1}{180} \cdot 10^{-4} \cdot \frac{99}{100}}{\frac{1}{5} \cdot 10^{-4} \cdot \frac{1}{100}} \\ &= \frac{\frac{99}{180} \cdot 10^{-6}}{\frac{1}{5} \cdot 10^{-6}} = \frac{\frac{99}{180}}{\frac{36}{180}} \\ &= \frac{99}{36} = \frac{11}{4} \end{aligned}$$

Consider the following function:

$$\mathbf{y} = \mathbf{x} \oslash \mathbf{v} + \mathbf{b}$$

Here x, y, v and b are vectors and $\oslash$ represents element-wise division.

This function is part of a larger computation graph, resulting in a scalar loss l. We want to implement this computation in an automatic differentiation (AD) system, as discussed in the lectures.

36. Work out the scalar derivative of y_i over x_j. Which is the correct solution?

A $\frac{\partial y_i}{\partial x_j} = \begin{cases} 1/v_j & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$ ✓

B $\frac{\partial y_i}{\partial x_j} = \begin{cases} 1/v_j & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$

C $\frac{\partial y_i}{\partial x_j} = \begin{cases} 1 + b_j & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$

D $\frac{\partial y_i}{\partial x_j} = \begin{cases} 1 + b_j & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}$

In scalar terms, the computation is $y_i = \frac{x_i}{v_i} + b_i$. When we work out $\frac{\partial y_i}{\partial x_j}$, we can tell immediately that y_i only contains x_j if $i = j$, so the derivative is 0 optherwise (leaving A and B as answers). If $i = j$, the the question reduces to

$$\frac{\partial y_j}{\partial x_j} = \frac{\partial x_j \frac{1}{v_j} + b_j}{\partial x_j} = \frac{1}{v_j} \; .$$

The AD system will give us a vector $\mathbf{y}^\nabla$, such that $y_i^\nabla = \frac{\partial l}{\partial y_i}$. Using $\mathbf{y}^\nabla$, we want to efficiently compute a vector $\mathbf{x}^\nabla$ such that $x_i^\nabla = \frac{\partial l}{\partial x_i}$.

37. Which operation computes this vector for us?

A $\mathbf{x}^\nabla = \mathbf{y}^\nabla \oslash \mathbf{v} + \mathbf{b}$

B $\mathbf{x}^\nabla = \mathbf{y}^\nabla \oslash \mathbf{v}$ ✓

C $\mathbf{x}^\nabla = \mathbf{y}^{\nabla \mathbf{T}} \mathbf{v} + \mathbf{b}$

D $\mathbf{x}^\nabla = \mathbf{y}^{\nabla \mathbf{T}} \mathbf{v}$

First, we work out x_i^∇.

$$x_i^\nabla = \frac{\partial l}{\partial x_i} = \sum_k \frac{\partial l}{\partial y_k} \frac{\partial y_k}{\partial x_i} = \sum_k y_k^\nabla \frac{\partial y_k}{\partial x_i} \; .$$

In the factor on the right, y_k only contains x_i if $k = i$. All other terms of the sum are zero, so we get

$$x_i^\nabla = y_i^\nabla \frac{\partial y_i^\nabla}{\partial x_i} = \frac{y_i^\nabla}{v_i}$$

where the last step fills in the answer to the previous question. This tells us that we are looking for a vector $\mathbf{x}^\nabla$ whose i-th element is given by y_i^∇ / v_i. We can compute such a vector as $\mathbf{x}^\nabla = \mathbf{y}^\nabla \oslash \mathbf{v}$.

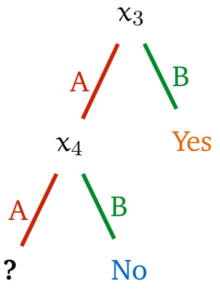
Consider the following task. The aim is to predict the class y from the binary features x_1, x_2, x_3 and x_4.

x_1	x_2	x_3	x_4	y
A	A	B	B	Yes
A	B	A	A	No
A	A	A	A	Yes
A	A	A	B	Yes
B	B	B	B	No
B	B	A	B	No
A	B	A	A	Yes
A	A	B	A	Yes
B	A	A	A	No
B	A	B	B	No
B	B	B	B	Yes
B	B	B	A	No

38. In standard decision tree learning (as explained in the lectures), without pruning. Which would be the first feature chosen for a split?
A x_1✓ **B** x_2 **C** x_3 **D** x_4

39. If we *remove* that feature from the data, which would be chosen instead?
A x_1 **B** x_2✓ **C** x_3 **D** x_4

Consider the following (partial) decision tree:



Note that this is just an arbitrary tree, not necessarily corresponding to the answers in the previous questions.

We can place either x_1 or x_2 on the open node (indicated by the question mark).

40. At this point, what is the information gain for x_2?
A -1 **B** 0✓ **C** 0.5 **D** 1

Thank you for your effort. Please check that you've put **your name and student number** on the answer sheet.