

CS392 Computer Learning : Exercises

Answers

1. (a) Describe the general batch learning problem, what are its main components?

This is the offline setting of learning, which assumes the existence of a training set, from which the learner generates a decision rule that it can use to classify new example.

A supervisor generates a training set of experience for the learner to learn from, and try to improve its performance at some task.

The main components of a learning system are:

- The input space X contains all possible unlabelled training and test examples.
- The label space Y contains all possible labels that can be given to all examples.
- The parameters space Λ which contains all possible parameters to the learning machine.
- The learning machine f .
- The loss function L chosen to assess the performance of the learning machine (eg. $L : Y^2 \rightarrow \mathbf{R}$).

- (b) Describe using notation the format of a training set.

The training set is usually a set of l examples which are themselves information pairs:

$$(x_1, y_1), (x_2, y_2) \dots, (x_l, y_l)$$

i. What is a feature vector?

x_i is a feature of the form $x_i = \begin{pmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^n \end{pmatrix}$ giving details of the n

features that describe that particular example. For example features for a diagnosis problem could be a set of symptoms or measurements taken from a patient.

ii. What are the labels in pattern recognition problems?

y_i is the associated label for each training example x_i .

In pattern recognition y_i is a finite discrete classification, such as 0 for benign, 1 for malignant.

iii. What are the labels in regression problems?

In regression problems y_i is a continuous real value, such as the price of a stock, or the height of a person.

iv. What is the main task of the learning machine?

Given a new test example x_{l+1} , can we predict its associated label y_{l+1} . Usually the supervisor holds back some of the training data (not including it for training) so that it can be used as a separate test set. The supervisor can then assess how accurately the learning machine is performing using this test

set by checking that the predictions given by the learning machine match that of the real labels.

Using the notation described earlier, the learning machine f , using parameters Λ should be able to map an unlabelled example to the correct label (eg. $f : X, \Lambda \rightarrow Y$).

- (c) Describe how a medical diagnosis problem (such as predicting cancer risk) could be formulated as a pattern recognition problem. Hint: describe what the training and test examples would be, in terms of relevant features and labels.

The training and test data would have to be gathered by a doctor, who will act as the supervisor of this task. Firstly the doctor would have to select some relevant features which he can measure from the patients.

The doctor of course cannot be certain which features are the most relevant. The doctor can be sure however that features such as the patients name, their telephone number, the type of car they drive, etc. will not help in predicting whether they are at risk of cancer, so can be ignored and not included as features. Perhaps more relevant feature measurements would be: smoker/non-smoker, age, sex, cancer in family, blood pressure, fitness level, alcohol consumption etc.

The classification labels, could represent the patients risk of cancer, eg. -1 no risk or +1 high risk. For the training data the doctor could take the measurements of the features for the patients that they treat, then over time add the corresponding labels retrospectively whether each of them developed cancer.

The doctor could then feed this training data into the learning machine. The learning machine would then generate a decision rule from this training data. The doctor could then use the learning machines decision rule to screen new test patients, by measuring their features and feeding them into the decision rule to get a

prediction about their risk of cancer.

To assess the performance of the learning machine, the doctor could keep back some the training data and use it as test data, so that they could compare the learning machines predictions with that of the real classifications.

- (d) Describe how we could predict the speed of a car, from its description by formulating the problem for regression analysis.
Hint: again consider features and labels.

As described earlier, the first task would be to pick some relevant features that would best help predict the speed of a car such as engine size, engine type, wheel size, weight of engine, automatic or manual transmission, weight of frame etc. We would not use features such as the number plate, the name of the owner, the colour of the upholstery etc.

The label could be a positive continuous real value specifying the corresponding time taken for that car to reach 60mph.

2. The following training set consists of noughts and crosses in the plane; the coordinates of the noughts are:

$$(0, -2), (0, -3), (1, -1), (1, -2), (1, -3), (2, 1)$$

and the coordinates of the crosses are:

$$(0, 1), (0, 2), (-1, 0), (-1, 1), (-1, 2), (-2, 0), (-2, 1)$$

- (a) i. Find the separating hyperplane for this data.

- ii. What does the separating hyperplane do?

The separating hyperplane divides the feature space into two distinct regions. Negative examples lie below the hyperplane, and positive examples are above the hyperplane.

- iii. What are the features in this data, how many features n are there?

In this example there are $n=2$ features, which describe each examples in a 2-dimensional plane.

iv. What are the labels y_i in this example ?

The labels in this problem are noughts \odot and crosses \times , however to solve this problem using an SVM we would have to encode each of these classes as ± 1 .

v. How many training examples l are there?

This problem has 13 training examples.

vi. What is the feature space?

In this problem the feature space is the space of all possible pairs of coordinates in the 2-dimensional plane.

(b) i. Which training examples are support vectors?

The support vectors for the cross coordinates are $(-1,0)$ and $(0,1)$, and for the nought coordinates $(0,-2)$, $(1,-1)$. Notice that these points lie on the margin hyperplane's.

- ii. In general what is the relationship between support vectors and the separating hyperplane?

The support vectors lie on the separating hyperplanes margins $H=\pm 1$. These training points can be thought of as defining the margin, being the most important examples that define the learning problem.

- (c) i. Find the predicted classifications (\times or \odot) for the new points

$$(-1, 3), (1, 0), (1, -4)$$

ii. Which points can we be more confident in predicting and why?

We can be more confident in predictions for $(-1,3)$ and $(1,-4)$ as these points lie outside the margins, however we cannot be as sure about the point $(1,0)$ as it lies within the margin.

3. (a) i. In support vector machine what is the equation for the separating hyperplane H ?

$H = (\mathbf{w} \cdot \mathbf{x}) + b = 0$ where w is the n dimensional weight vector

$$\mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} \text{ and } b \text{ is the bias, that define the hyperplane.}$$

Often we introduce b as an extra weight vector such as $w_{n+1} = b$, and then augment the training vectors with an extra redundant feature $x_{n+1} = 1$.

- ii. Why do we try to find the optimal separating hyperplane for a set of training data?

To hopefully improve generalisation over new training examples.

iii. What does the separating hyperplane represent?

The separating hyperplane represents the decision rule learnt by the SVM.

iv. What is the margin δ of a hyperplane.

The margin δ is the distance of the closest training examples to the hyperplane $H = 0$.

v. What are the support vectors?

These are the points that lie on the parallel margin hyperplane's $H = \pm 1$. They have distance δ from $H = 0$.

(b) Describe the difference between the separable and the non-separable cases of pattern recognition using support vector machines.

In the non-separable case, no separating hyperplane exists that can separate positive and negative examples. For the non separable case we must introduce slack into our constraints to allow overlap. The problem then becomes to minimise this overlap.

(c) i. Define the optimisation problem, algebraically (eg. main

optimisation and side constraints), for the SVM in the separable case.

The optimisation problem is:

$$\begin{aligned} & \text{Minimise} \\ & \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) \\ & \text{Subject to the constraints} \\ & y_i(\mathbf{w} \cdot x_i + b) \geq 1 \text{ for } i = 1, \dots, l \end{aligned}$$

ii. What is the intuition behind minimising the weights?

The distance of an example x_i from the hyperplane is

$$d(x_i, H) = \frac{(\mathbf{w} \cdot x_i)}{\|\mathbf{w}\|}.$$

Therefore if we minimise the weights \mathbf{w} then we are minimising $\|\mathbf{w}\|$ which is maximising the margin.

iii. What is the intuition behind the side constraints?

They make sure that the positive examples ($y_i = 1$), lie above hyperplane $H = 1$, and negative examples ($y_i = -1$) lie below hyperplane $H = -1$. For example:

$$y_i = 1 \text{ gives } \mathbf{w} \cdot x_i + b \geq 1 \Rightarrow x_i \text{ is above or on } H = 1.$$

$$y_i = -1 \text{ gives } -(\mathbf{w} \cdot x_i + b) \geq 1 \Rightarrow (\mathbf{w} \cdot x_i + b) \leq -1 \Rightarrow x_i \text{ is below or on } H = -1.$$

(d) i. Define the optimisation problem, algebraically (eg. main optimisation and side constraints), for the SVM in the non-separable case.

In the non-separable case there does not exist a hyperplane that can separate positive and negative examples.

The optimisation problem is:

$$\begin{aligned} & \text{Minimise} \\ & \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) + C(\sum_{i=1}^l \xi_i) \\ & \text{Subject to the constraints} \\ & y_i(\mathbf{w} \cdot x_i + b) \geq 1 - \xi_i \text{ for } i = 1, \dots, l \end{aligned}$$

The term C determines how much slack we allow in the constraints.

ii. What is the functionality of the slack variables?

We introduce slack variables ξ_i for each of the training examples x_i , allowing it to cope with non separable data. The ξ_i slack variables soak up the errors on the wrong sides of the hyperplanes $H = \pm 1$.

4. You are given the training set,

$$x_1 = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 1 & 1 \end{pmatrix}, y_1 = 0,$$

$$x_2 = \begin{pmatrix} 0 & -1 & 0 & 1 & -1 & 0 & 1 \end{pmatrix}, y_2 = 1$$

and the new unlabelled test example

$$x_3 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

- (a) i. State the least-squares optimisation problem algebraically.

Minimise

$$\sum_{i=1}^l (y_i - (\mathbf{w} \cdot x_i + b))^2$$

This is essentially minimising the sum of the square errors over the training set. You may have come across this in your neural networks course, as the foundations for the back propagation algorithm.

- ii. What is the matrix formulae for solving least-squares regression in primal form?

The solution for the optimal weights \mathbf{w} is:

$$\mathbf{w} = (Z'Z)^{-1}Z'y$$

To get the prediction of the label \hat{y} of a new test example x_i we must create an augmented vector z with the same features as x_i but with an extra redundant feature $x_i^0 = 1$ (explained below).

Now to get the prediction we must calculate:

$$\hat{y} = (\mathbf{w} \cdot z)$$

A more in depth description of how to create these matrices is given below.

As mentioned earlier we think of the training set as a sequence of l examples, represented as information pairs

$(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)$. Each example contains an n

dimensional feature vector $x_i = \begin{pmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^n \end{pmatrix}$ (which describes each

example), with a corresponding label y_i (which is the target to be learnt). With regression problems y is a continuous real value.

The matrix Z mentioned in the least squares formula (??) above is of the form:

$$\begin{pmatrix} 1 & x_1^1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2^1 & x_2^2 & \cdots & x_2^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_l^1 & x_l^2 & \cdots & x_l^n \end{pmatrix} \quad (1)$$

The matrix Z is an $l \times (n+1)$ matrix, which defines the training set feature vectors as its rows, augmented with an extra column of 1's to add an extra constant term w_0 (sometimes known as the bias b) to the solution found by the ridge regression algorithm. This means that the resultant solution of optimal

weights is an $(n+1) \times 1$ vector $w_{opt} = \begin{pmatrix} w^0 \\ w^1 \\ w^2 \\ \vdots \\ w^n \end{pmatrix}$.

- iii. What is the matrix formulae for solving least-squares regression in dual form? Hint: take out the regularisation term from dual ridge regression in your notes.

$$\hat{y} = y'(ZZ')^{-1}Zz'$$

iv. In the dual form can we explicitly find the weight vector \mathbf{w} ?

If we do not use a kernel to map into a higher dimensional feature space then we can derive the weights back into a weight vector. However with using the dual you must remember that you are in fact finding a solution in terms of l dual variables α_i , and not the $n + 1$ dimensional weight vector. To make a prediction you do not need to explicitly use the weight vector, because it is implicitly defined by the dual variables.

- (b) Find the least-squares prediction of the unseen label y_3 using either the primary or the dual variant of the least-squares method. Hint: you may need to use this rule to invert a 2×2 matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

- (c) Justify your choice of method (i.e. primary or dual form). Could you have used the other method?

You should have used the dual method, because with $(n = 7) > (l = 2)$, so for the primal you would have to find the inverse of an 8×8 matrix. However with the dual version you only need to invert a 2×2 matrix.

- (d) i. State the ridge regression optimisation problem algebraically.

Minimise

$$C(\mathbf{w} \cdot \mathbf{w}) + \sum_{i=1}^l (y_i - (\mathbf{w} \cdot x_i + b))^2$$

- ii. How does this differ to the least-squares optimisation problem?

The regularisation term is $C(\mathbf{w} \cdot \mathbf{w})$ is added to the optimisation problem. This will now keep the weights small and minimise the sum of the square errors over the training set. This is like trying to use the least amount of ‘energy’ to find a solution. The larger C is the more emphasis there is on minimising the weights, and less on minimising the square errors.

- iii. How could this modification improve the learning process cope with noisy data?

By minimising the weights we try to find the simplest hypothesis as possible. If the training data contains a certain

level of noise (such as human error on input, or the example is misclassified), then when the learning algorithm tries to find a hypothesis that fits this training data precisely then it will overaccomodate for these extreme/error examples (aka over-fitting). This will then mean that the learning machines decision rule will not generalise well for new unseen test examples.

By adding this regularisation term to limit the complexity of the decision rule created by the learning machine, we can stop it from overfitting the training data and hopefully generalise better.