EE2211 Finals

Used for overview of the 500 slides. Detailed notes are written in the combined slides themselves.

Always remember the workflow:

1. What is the objective/problem?
2. Is it appropriate for ML (learnable from data?)?
3. What are the inputs and outputs (features and labels to predict for if any)?
4. IDENTIFY THE TASK
    1. Regression?
    2. (Binary/Multi-class) classification?
    3. Clustering?
5. Decide on the technique/solution (for my own vscode):
    1. linear regression
        1. regression task and binary classification task (y = -1 or 1)
    2. linear regression with auto one-hot classification
        1. multi-class classification
    3. polynomial regression - input dimension more than one
        1. regression task, binary classification and multi-class classification (manual one hot)
    4. ridge regression - X.T X is not invertible or prevent overfit
        1. regression task, binary classification and multi-class classification (manual one hot)
    5. ridge polynomial regression
        1. regression task, binary classification and multi-class classification (manual one hot)
    6. Classification Tree
        1. Classification task (predict discrete variables) - minimising impurity
    7. Regression Tree
        1. Regression task (predict continuous variables) - minimising MSE

Ch1-3

Chapter 1: Intro (Step 1: Problem Definition)

4

A program is said to learn from experience E - with respect to some class of tasks T - and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. (perhaps test examples)

5 6

- Output also called label
- Data output pair
- the function is said to be learned

7
Venn diagram of AI > ML > Deep Learning

8
ML is not always useful; two scenarios to use

9-13
applications of ML

14 (very important)
summary of types of ML and each of their characteristics
workflow of ML and **problem definition**

15-17
supervised learning and TF statements about the two TASKS

- regression TASK (if y is a continuous output) - slide 16
  - learn a function f(x) to predict **real valued** y given x
  - techniques include: linear, polynomial, ridge regression (k-NN also can)
  - evaluation metrics include mean squared error and mean absolute error
- classification TASK (if y is a discrete/categorical output) - slide 17
  - learn a function f(x) to predict **categorical** y given x
  - **note that (linear/poly/logistic) regression CAN be used for classification as they are simply techniques, in fact logistic regression is used for classification task only**
  - polynomial (linear) regression used for binary classification and multi-category classification, using sign and one hot encoding
  - logistic regression also used for binary and multi-category classification, but using soft threshold function
  - k-nearest neighbour classification after feature extraction

18-19
unsupervised learning (unlike supervised, no label given)

- clustering TASK - slide 19
  - no expected outputs, but more to find underlying patterns and hidden structure in a given sample data set
  - techniques used: K-means clustering, fuzzy c-means clustering

20-22
reinforcement learning - slide 20

- given seq of states S and actions A with (delayed) rewards R, output a policy (Product (a,s)) to guide what actions a to take in state s
- in other words, after learning, **given a state, what is the best action to take**

23
good example of the difference between the 3

24-29
feature (attributes of sample) extraction - for the model to look out for (colour, shape)
decide labels to classify the data into - for the model to label each data (correct shape, wrong shape)
feature space (26)
nearest neighbour classifier and how it works (29)

30-31
"inductive" vs "deductive" learning: probability-based vs rule-based/deterministic

32
summary of chapter 1

33

> nearest neighbour classifier is supervised, inductive and not a feature selection

Chapter 2: Data engineering and processing (Step 2: Data Preparation)

37
what are types of data

38
ways to view data

39-47
"level/scales of measurement": NOIR (39 and 46 best)
example on slide 67: "practice question"
"nominal": named - slide 41
"ordinal" : named + ordered - slide 42
"interval": named + ordered + equal interval - slide 43
"ratio": named + ordered + equal interval + has "true" zero - slide 44

48-49:
"numerical" - "quantitative" - Interval, Ratio
"categorical" - "qualitative" - Nominal, Ordinal

50-51:
"missing data": "NA " with space

52-54:
"data wrangling": raw to a more useful format before ML

55
"formatting data" "one-hot encoding": unify entities within a vector
"normalization": linear scaling and z-score standardization
why is normalisation important: so that data of large value and high variance from others are normalised, all data/features are on comparable scales and no feature completely dominate the feature space.

56-58
"data cleaning": detect outliers/missing data and correct/remove
data imputation

59-65
"data integrity" - slide 60
visualisation of data
"boxplots": - slide 64

Chapter 3: Linear Algebra (MA1508), Probability and Statistics (GEA1000 and JC Stats)

75
"a set is an unordered collection of unique elements"

76
"capital sigma" for summation
"capital pi" for product

77-79 (refer to matlab/MA1508 notes)
linear independence
full rank: "if all rows or columns" of a square matrix X are linearly independent, X is invertible
check invertible by `det(A)` or `rref(A)` or any of the equivalent statements
"rank(X)" is the max no of linearly independent col/row of X

- once again just `rref(A)`

81-90 (refer to GEA1000 cheatsheet)
"casuality" "Randomized Controlled Trial" - 84
"correlation" - slide 86, 87"pearson coefficient"

- r is strength and sign (recall GEA)
  "simpson's paradox": small trends reverse or disappear in big - slide 89

91 - 109 (Refer to JC stats or GEA1000)
outcomes (not decomposable) vs events (specific group of possible outcomes, hence decomposable and subset of the sample space)
"Axioms of probability" - slide 93
"random variables: discrete and continuous" - slide 94, 96 - 98, 99 - 101
"two basic rules" - slide 105
"bayes' rule" - slide 106 - base rate fallacy (conditional probability)

Ch4-6

Chapter 4 Linear Algebra, Sets and Functions

8:
"dot product or inner product"
x dot y = x.T@y (matrix multiply)
Geometric rep: **length** (hence scalar) of projection of x on y (i.e. how much one vector extends in the direction of another vector)

9-22 (Operations on vectors and matrices - basic stuff)
"matrix-vector product"

- Wx = gives column vector
  - if W *x is of (m x n)* (n x1)
  - result is a (m x 1) column vector
    "vector-matrix product"
- x^TW = gives a row vector
  - if x^TW is of (1 x m) * (m x n)
  - result is a (1 x n) row vector
    "matrix-matrix product"
- always row x col
- (row 1 x col 1) * (row 2 x col 2)
  - product will be outer rainbow (row 1 x col 2)
  - can multiply only if inner rainbow col 1 == row 2
    inverse must be invertible and square (refer to equivalent statements)
- use inv in matlab to check
  "adjugate or adjoint of A"; "determination computation"
- transpose of cofactor matrix of A = `C'`
- just use matlab `cofactor()`
  - `C = cofactor(A) returns the matrix of cofactors of A.`
  - `If A is invertible, then inv(A) = C' / det(A).`
  - `C = cofactor(A, i, j) returns the cofactor of row i, column j of A.`
    - the T shape thing

23-35 (Systems of Linear Equations) - very important
- X is the input data matrix (row = no of eq, col = no of unknowns) - this is the features of samples
- eq > unknowns = over determined
- eq < unknowns = under determined
- eq == unknowns = even determined
- w is the weights (col vector)
- y is the output () - this is the labels of output by model

w^T X = y^T == X^T w = y

- so eq and variables in X are now swapped where col rep eq and row rep unknowns

26
Over determined, under determined, even determined
we are always trying to solve for w because we want to find the coefficients of the system of equations such that all the variable values in X get give y using the weights w

27
even determined - one unique solution
w = inv(X)@y given that X is invertible or else NO SOLUTION

29
over determined - no exact solution (unless rank(X) = rank(`[X y]`))
(least squared) approximated solution using left inverse
w = leftinv(X)@ y = `[ inv(X.T @ X) @ X.T ]` @ y
of course given X.T @ X is invertible, or else NO SOLUTION

31
under determined - infinite solutions (in terms of one or more of the unknowns)
a unique solution is possible by constraining the search using w = X.T @a, where a = XXT^-1 y
no solutions if system is inconsistent (rank(X) < rank(`[X y])) (constrained solution) w = rightinv(X)@ y = [ X.T @ inv(X @ X.T)]` @ y`
of course given that X@X.t is invertible or else NO SOLUTION

35 SUMMARY

36-47 - Set and Functions
37
"Notations: set"

39-42
"functions"

- scalar function can have vector argument
- vector function returns a vector and can have a scalar input argument

- "$f:\mathcal{R}$" ^ d -> R to find mapping "real d-vectors to real numbers"
- inner product function

43-45
"linear functions" just mean it can be expressed as a linear combination of vectors

46
"affine functions" is linear combination + a constant (of so called offset/bias)

52 is a very good summary for linear functions, inner product function and affine function

Chapter 5 Linear Regression for regression task

56
Differentiation of a scalar function wrt a vector

57
differentiation of a vector result is hxd

58
4 useful vector-matrix differentiation formulae to use immediately

59-77
"linear regression"

- when output can be modelled linearly, where y = Xw + b
- x is an instance of a data, represented as a column vector. rows represent the features.
- for each of these x, there is a corresponding y output.
- y = x^T w + b (transpose x due to lin alg mechanism only)
- if there are multiple data (x, y) data points, it will be Xw = y
    - X is the stacked transposed x vectors. Hence, columns now represent features, rows represents number of such instance
- This is a minimisation problem, hence cost function is the objective function here, which is the sum of the loss function. Loss function is the error output by model compared to actual y. And we want this to be minimum for best model
    - first step is to train using input X matrix and correct y output vector (set them up)
    - solve for optimal w that minimises the cost function
        - using vector-matrix notation, error = Xw - y, for some w. Xw is the output after training. y is the actual output
        - rewrite the cost function into the compact form $L(w) = \| Xw - y \|^2 = (Xw - y)^T(Xw - y)$, which by taking derivative of L wrt w and setting derivative to 0 to find global minimum, we will get the expression $X^TXw = X^Ty$
        - hence, we solve w by $\mathbf{W}=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ (or the other equivalents)
    - what are the bias (rows of 1s)in matrix X

        - bias is an extra constant (the intercept) added to the model to allow the model to make non-zero predictions when all inputs are zero
            - $y = Xw + b$ for a full linear model, where b is the bias
            - another way is to add a column of ones in X
- "Learning of Vectored Function (Multiple Outputs)"
    - if y has 2 columns, you are just doing 2 sets of weights (2 col also) and they are independent of each other. Just using the same training X

read the output W:

- first row is always for the bias term
- if multiple column, they correspond to the multiple column output respectively

Chapter 6 Regression for classification and ridge + poly regressions

80 - 89

- "linear regression (for classification)"
    - recall that X are the features and y are the labels
    - "binary classification"
        - assign one class to positive and the other to negative
    - "one-hot encoding/assignment"
        - a way to represent categorical variables as numeric vectors (decoder type stuff)
        - Class 0, Class 1, Class 2 for the different categories
        - **For each row of Y, the column position of the largest number (across all columns for that row) determines the class label.**

90

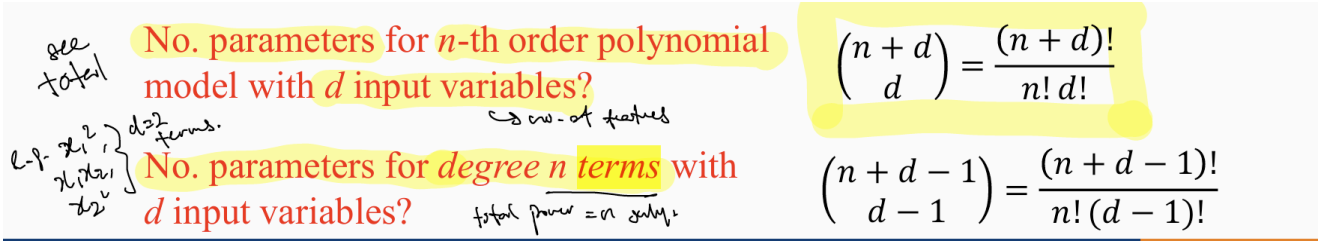
- "ridge regression"
    - shrink the regression coefficients w by imposing a penalty on their size, to prevent overfitting and handle issues like multicollinearity or non-invertible $X^T X$
        - w (weights) tells how much each feature influences the prediction, without penalty like in least squares, the goal is to only pick weights that minimise the squared error on training data, but if there are many features or features are highly correlated, the model will pick very large values of w to fit the training data perfectly.
        - large weights mean the model is very sensitive: small changes in the feature cause big jumps in predictions. This is an undesirable overfitting where the model fits noise and performs poorly on new data
        - in ridge regression, by adding the last term, we pay a cost for large coefficients. To keep the cost low, the model is "encouraged" to keep the

weights small = smoother and more stable models that are less sensitive to noise

- The **loss function** for ridge regression is: $L(w) = ||Xw - y||^2 + \lambda||w||^2$ where:
  - $||Xw - y||^2$ = sum of squared errors (same as OLS)
  - $||w||^2 = w^T w$ = squared L2 norm of the weights
  - $\lambda \geq 0$ = regularization parameter controlling the amount of shrinkage
- solution using a linear model - slide 91
- primal form is for overdetermined - slide 92
- dual form is for underdetermined - slide 93

Aspect	Ordinary Least Squares	Ridge Regression
Objective	Minimize ($||Xw - y||^2$)	Minimize ($||Xw - y||^2 + \lambda||w||^2$)
Solution	($w = (X^T X)^{-1} X^T y$) (if invertible)	($w = (X^T X + \lambda I)^{-1} X^T y$)
Regularization	None	Yes
Use cases	Well-conditioned (X) matrices	Ill-conditioned (X), multicollinearity, high-dimensional data

- "polynomial regression": - 94
  - fit a polynomial function of degree n for the data points
  - how many terms using nCr
    - k = number of variables
    - d = degree of polynomial
    - (k+d)C(d) for TOTAL number of terms
    - to find for specific level d, do (k+d)C(d) - (k + d-1)C(d-1)



Ch7-9

Chapter 7: Over-fitting, feature selection, regularisation, bias/variance trade-off

9

- zero overlap between training and test sets
- goal of regression is prediction on new unseen data, i,e, test set, which is important for evaluation

11 12

- shows using order 9 has overfitted using training samples

14 15

- underfitting using order 1

16 17

- good fit

19
Overfitting definition and reasons and solutions

20
Underfitting definition and reasons and solutions

21
summary of how error, model complexity, number of features are linked, and how underfitting and ovefitting fits in the graph

23 24 25
feature selection using pearson's correlation r

26 - 31
Regularisation

32-42
Bias-Variance definitions and Trade off

Chapter 8: Optimisation and Gradient Descent and different learning models

51
review of how to write cost function, loss function, learning model, and regularisation into one statement

52
definition of each of the terms mentioned above

53 54 55
Gradient descent motivation, pseudo code

56 - 74
picking learning rate will affect how fast and whether convergence is possible

- just nice = rapid convergence
- too small - slow convergence

- too big = overshoot local minimum (slow convergence) or may never converge

75-80
learning models that we use for ML reflects our own belief about the relationship between the features and target.

- sigmoid function
- ReLU function

instead of square error losses (good for regression), for classification, binary (cross-entropy) loss function

Chapter 9: Decision Trees, Random Forest

86:
Decision Tree Classification

88:
Basic terminologies are the same as those for CS trees: branch, parent node, children node, terminal node/leaf, decision node

89:
considerations when building a classification decision tree

the decision tree MODEL is modeled by all the decision rules at each decision node, since that is how the model classify a given test data starting from root, goes through all the decision nodes, then land on a leaf node, which is considered classified.
every split is a component of the weight vector. Each split is by a decision rule which was trained by comparing all features and threshold and computing each impurity before determining the exact decision rule that minimises the impurity. All these decision rules at each split makes up the typical weight vector

cannot too large (too complex) or else overfit but also impossible to find smallest tree

90 91
Node impurity for classification tree and 3 types of measure (AND a summary)

92
Gini Impurity:

- using proportion of objects by class to compute using a formula the pureness in each node
- Overall Gini should decrease every depth to ensure decision tree is making progress
    - it is just the weighted average of the nodes's Gini

93
Entropy (quite similar to Gini)

94:
Misclassification. unlike Gini and Entropy, only look at most common class.
quick trick is Overall misclass is also total misclassified samples at that depth/total sample data

95
Classification tree learning pseudo code

97
Advantages and disadvantages of classification decision tree

main advantage is it does not assume any relationship
main disadvantage is it is easy to overfit and each decision affects the next

98
How to reduce overfitting of tree

99
Regression Tree (seeks to predict continuous variables - regression task) unlike classification trees which seek to predict discrete variables (or classify things into classes - classification task)

predict ave value instead of picking majority
use MSE instead of impurity
total MSE

101 - 111
regression tree example

112
How to reduce instability - using a random forest instead of a single decision tree
trees have low bias but high variance
can mitigate high variance by simulating noise in training data and do 100 trees, then take the average or most frequent class predictions, lets say over 100 trees (a random forest)

113
a way to perturb data is using bootstrapping and what are its characteristics

Random forest should have similar bias to single decision tree, but much lower variance

Ch10-12

3:
summary page and learning outcomes based on chapters

Chapter 10: Data Partitioning, Cross Validation, Test Performance Evaluation Metrics

7
hyperparameter

9
training data set: train the ML model
validation data set: choose the parameter or model (run variety, choose lowest validation error)
test data set: evaluate the real performance and generalisation of final trained model

11
how to use validation accuracy

13-16
kfold cross validation: partition into k parts and run k times to use each part as validation (remainder as training)
4 fold cross validation

17
take parameter/model with best average validation performance over k folds

19
after doing validation performance (cross validation) to select parameters, perform test performance
test performance shows how well our model generalises

20:
validaiton is used only when you need to pick parameters or models
can just partition data into only training and test set

22:
evaluation metrics for regression tasks: Mean Square Error and Mean Absolute Error

23:
evaluation metrics for classification tasks

Validation of data

Priority of accuracy

1. Mamimise accuracy of validation set (unseen data)
2. (Tie-breaker) Maximise accuracy of training set (Prevent overfitting)

Confusion Matrix

CM	Predict $\hat{P}_1$	Predict $\hat{P}_2$
Actual P_1	TP	FN (Type 2 Error)
Actual P_2	FP (Type 1 Error)	TN

Error	Interpretation
Type 1	False Positive / False Alarm / Overestimation
Type 2	False Negative / Miss / Underestimation

Imbalanced Classification

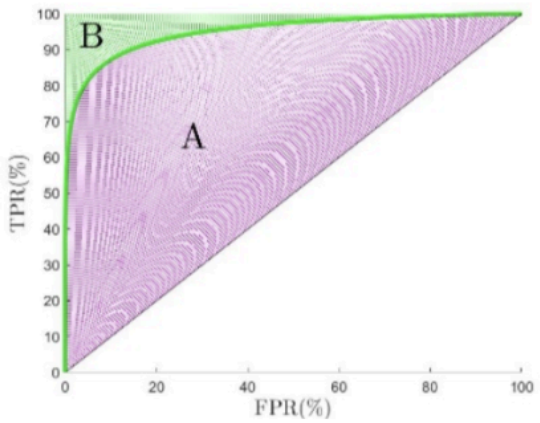
Typically

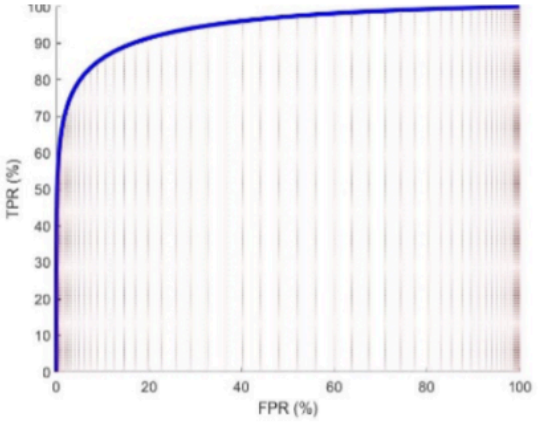
- Majority class $\Rightarrow$ Label 0 (Negative)
- Minority class $\Rightarrow$ Label 1 (Positive)

Formulas

- Recall / Sensitivity / Hit Rate / True Positive Rate (TPR) $= \frac{TP}{TP+FN} = 1 - FNR$
- Miss Rate / False Negative Rate (FNR) $= \frac{FN}{TP+FN} = 1 - TPR$
- Selectivity / Specificity / True Negative Rate (TNR) $= \frac{TN}{TN+FP} = 1 - FPR$
- Fall-out / False Positive Rate $= \frac{FP}{TN+FP} = 1 - TNR$
- Precision / Positive Predictive Value (PPV) $= \frac{TP}{TP+FP} = 1 - FDR$
- False Discovery Rate (FDR) $= \frac{FP}{TP+FP} = 1 - PPV$
- False Omission Rate (FOR) $= \frac{FN}{FN+TN} = 1 - NPV$
- Negative Predictive Value (NPV) $= \frac{TN}{FN+TN} = 1 - FOR$
- Classification Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$

Visualising Metrics





Gini Coefficient Curve

Area of $A + B = \frac{1}{2} \times$ Area of whole plot

Gini $G = \frac{A}{A+B}$

Receiver Operating Characteristic (ROC) Curve

Shows the boundary between A and B from Gini Coefficient Curve

$AUC = (\frac{1}{2} \times$ Area of whole plot $) + A$

Assume Area = 1, then $A = AUC - \frac{1}{2}$

Relationship between Gini and AUC

$G = 2 \times AUC - 1$

25- 26:
evaluation metrics for classification tasks:

- confusion matrix and cost matrix (binary classification)
  27:
  why cost matrix is useful (only focus on those that we really want to reduce)
- usually true positive and true negative is not an error (we dont need to reduce that) so = 0

28:
for unbalanced data, instead of looking at total cost/accuracy

- can focus on Recall and Precision instead

33
changing threshold can change TP FP FN TN

33:
evaluation metrics for classification tasks:

- confusion matrix and cost matrix (multicategory classification)

Chapter 11: K-means clustering

40:

- we do not always have labeled data (no expected y outputs)

41:
evaluation of unsupervised learning is hard as there is no reference set of data to validate or test

42:
types of unsupervised learning techniques

43 - 60
illustration of kmeans clustering (2D)

61:
Basic/Naive way of k means

64 - 65:
Optimisation Objective function (within-cluster variance) for k-means

66
Math behind Assignment and Update step

67:
reasoning behind why total loss will not increase with those two steps

68-70:
different initialisations give different clusters

71:
k means is not guaranteed to find a global minimum, it only finds local minimum

72:
different ways to initialise

73:
hard/soft clustering
fuzzy clustering uses probability

74-79
fuzzy k-means

Chapter 12: Neural Networks (NN)

86
perceptron

87-88
activation functions
sigmoid (asymptote 0 to 1 asymptote)
relu: e.g. max(0,a) means if a < 0, return 0, else return a

89
important to know the dimension of Weights vector

90:
things to note for MLP

91:
nested function

93:
backpropagate to update w

94:
training: forward and backward
why backward? when training, we need to use y_expected and x_expected.
the true reference point is at the output (last layer) which is our y_expected. x_expected is from previous layer. and x_expected now is previous layer's y_output, Hence, train w by propagating backwards.
hidden layers

95:
backpropagation is gradient descent

96:
testing: forward
simply put in a test data and run through the MLP

100-110:
CNN