1 Recap of supervised learning concepts

Terminology

1. Give a definition of 'over-fitting'.

   Solution: *Overfitting is where a model fits the training data too well, capturing noise in the data rather than the true function. This can lead to a model that performs well on training data but fails to generalize to unseen data.*

2. Can over-fitting occur even if there is no noise in the data?

   Solution: *Yes. Overfitting is not solely dependent on the presence of noise; it can also happen when a model is excessively complex relative to the size of the training data. For example in the lecture I showed a model fitting a sine wave. Even if there was no noise, the small training data sample might not capture all the parts of the function, and the model may interpolate in incorrectly in those regions.*

3. What are the parameters controlling model fit in:

   (a) a k-nearest neighbour model

   Solution: *Primarily k, the number of neighbours. However, the choice of distance measure (not necessarily Euclidean distance) can also have a significant impact.*

   (b) a decision tree

   Solution: *Primarily the tree depth, but also others like the choice of node-splitting criterion.*

   (c) a logistic regression

   Solution: *In a plain logistic regression there are no parameters — it has a unique global minimum. However we can regularize (constrain) weights, with L2 regularization. This makes a simpler model, since the weights are constrained in a smaller 'ball', and less likely to grow large for particular features in the data (i.e. overfit).*

   (d) a neural network

   Solution: *Many. Choice of learning algorithm. Number/size of layers. Any weight regularization, like L2 or dropout. Choice of activation functions. Number of epochs to train. Batch size. The list goes on.*

4. If I increase the value of k in a k-nn model, the decision boundary gets simpler. True or false? Why?

   Solution: *True. The model takes more of the training data into account, thus smoothing out any random fluctuations. Eventually the decision becomes constant for any input.*

5. If I hold k constant, but increase the amount of training data, what happens to (a) the time it takes to train, (b) the time it takes to make a prediction, (c) the generalization error?

   Solution: *Training time is constant since k-nn is a memory based algorithm. The Testing time will increase as there are more points to search through when finding the neighbours. Generalization error will most likely decrease, but it depends on the quality of the new training data, i.e., whether it is noisy or not.*

Cross-entropy loss

1. A binary logistic regression predicts the probability of $y = 1$. I give you an input $\boldsymbol{x}$, where label is $y = 1$. My model returns $f(\boldsymbol{x}) = 0.7$. Calculate the cross-entropy loss. Remember to use the **natural** logarithm.

   Solution: *The cross entropy is* $\ell(y, f) = -(y \ln f(\boldsymbol{x}) + (1 - y)\ln(1 - f(\boldsymbol{x})))$.

   If $y = 1$, *this reduces to* $\ell(y, f) = -\ln f(\boldsymbol{x}) = -\ln 0.7 \approx 0.3566$

2. Now, the true label is $y = 0$. Your model returns $f(\boldsymbol{x}) = 0.2$. Calculate the cross-entropy loss.

   Solution: $-\ln(1 - 0.2) = -\ln 0.8 \approx 0.2231$

3. Now, I give you an example $\boldsymbol{x}$, where $\boldsymbol{y} = [1, 0, 0]$, i.e. a multi-class problem. I build a new model, and it predicts $f(\boldsymbol{x}) = [0.6, 0.3, 0.1]$. Calculate the cross-entropy loss again.

Solution: $-\ln 0.6 \approx 0.5108$

4. Sketch a probability simplex and indicate roughly where this multi-class model predicts.

Solution: *Simplex with marker in the class 1 area.*

Gradient descent.

1. Given a loss function $\ell(y, f) = (y - f(\boldsymbol{x}))^2$, where $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$, compute the gradient of ℓ with respect to a single element of $\boldsymbol{w}$, which we denote w_i.

Solution: $\frac{\partial \ell(y,f)}{\partial w_i} = 2x_i(\boldsymbol{w}^T\boldsymbol{x} - y)$

2. I am performing gradient descent on a large model using the cross-entropy loss, but the model is training too slowly and I believe overfitting. To resolve this I increase the learning rate to infinity, and remove all regularization. What's wrong with my reasoning?

Solution: *Learning rate infinity is nonsense. It needs to be a small constant usually close to 0. Removing regularization would cause more overfitting.*

3. What does the learning rate determine in gradient descent?
A. The size of each step during optimization.
B. The maximum number of epochs.
C. The global minimum of the function.
D. The complexity of the model.

Solution: *A*

4. What does the term "batch" refer to in the context of gradient descent algorithms?
A. The number of features in the dataset.
B. The entire dataset used for a single update of the model parameters.
C. The learning rate used in optimization.
D. The number of iterations required for convergence.

Solution: *B*

2 Empirical Risk Minimization

1. I want a neural net to classify images of people as 'happy', 'sad', or 'angry'. I'd like it to predict the probability of each class. The images are greyscale, resolution 256×256 pixels. What are the input ($\mathcal{X}$) and output ($\mathcal{Y}$) spaces, for this problem? What is the dimension of each, i.e. d, k?

   Solution: $\mathcal{X} = [0, 255]^{256\times256}$*, so* $d = 65,536$*, and* $\mathcal{Y}$ *is the 3-simplex, i.e. a distribution over* $k = 3$ *classes.*

2. What is the primary goal of empirical risk minimization in machine learning?

   Solution: *To reduce generalization error. Whilst we attempt to reduce empirical risk, it is only a proxy for the population risk.*

3. How does empirical risk differ from population risk in the context of ERM?

   Solution: *Empirical risk is a finite sample estimate of the population risk.*

4. Can you explain the relationship between the empirical risk and the training dataset?

   Solution: *The empirical risk is a quantity that could be calculated using the training dataset, but it could also be calculated using any other dataset.*

5. How does regularization contribute to the empirical risk minimization process?

   Solution: *It constrains the size of the hypothesis space / function class, thus making the estimation error smaller.*

6. List some examples of loss functions used in ERM.

   Solution: *Squared, cross entropy, 0/1 loss, margin losses (e.g. hinge loss in SVM), but also less well-known ones such as Huber Loss, which is less sensitive to outliers in regression tasks. There are many.*

7. Give a formal definition of the Bayes model for squared loss. Try to prove this yourself.

   Solution: $y^* = \mathrm{argmin}_f \mathbb{E}_{\mathbf{x}y}[\frac{1}{2}(f(\mathbf{x}) - y)^2] = \mathbb{E}_{y|\mathbf{x}}[y]$. *Either with or without the constant* $\frac{1}{2}$ *is fine.*

8. $R(f_{erm})$ is always greater than or equal to $R(f^*)$. True or false? Justify your answer.

   Solution: *True. The latter is defined via the minimiser of the population risk, within the model family* $\mathcal{F}$*, hence by definition is the smallest possible population risk given our model famiy.*

9. The empirical risk of the Bayes model, $\hat{R}(y^*, \mathcal{S}_n)$, can be either larger or smaller than $R(y^*)$, depending on the size of n. True or false? Justify your answer.

   Solution: *True. The empirical risk can either over or under-estimate the population risk, for any given model, including the Bayes model.*

10. $R(y^*)$ is always less than or equal to $R(f_{erm})$. True or false? Justify your answer.

    Solution: *True.* $R(y^*)$ *is the population risk of the Bayes model, hence by definition is the minimizer of the population risk, whereas* $R(f_{erm})$ *is the population risk of the ERM.*

11. The approximation error is due to the limited training data size. True or false? Justify your answer.

    Solution: *False. This is the definition of estimation error.*

12. Overfitting is said to occur when approximation error is too large, and estimation error shrinks. True or false? Why?

    Solution: *False. This is under-fitting. Approximation error corresponds to the ability of the model to fit complex functions, hence if it is very large, the model must be too simple for the problem at hand, and is thus unable to fit well... hence under-fitting.*

3 Generalisation bounds

Assume you have d features, and a model $f(\mathbf{x}) = \boldsymbol{w}^T\boldsymbol{x}$, but imagine that you are restricted to k bit integers for any of the weights. This is an example of an active area of research called "quantized Machine Learning".

1. What is the size of the function class here?

   Solution: *Every weight is restricted to be a k bit integer, we have 2^k options for each coordinate of the weight vector. Hence there are exactly $2^k \times \ldots \mathrm{d-times} \times 2^k = 2^{kd}$ options for the weight vector. So empirical risk minimization in this setup is a search inside a set of 2^{kd} linear functions, which is the size of the function class.*

2. If we were to construct a generalisation bound as we did in class, what does it represent?

   A) An upper bound on the empirical risk that holds with a certain probability

   B) An lower bound on the population risk that holds is guaranteed to hold

   C) An upper bound on the number of training examples that is guaranteed to hold

   D) An upper bound on the population risk that holds with a certain probability

   Solution: *D*

3. Consider a finite hypothesis class $\mathcal{F}$, where $|\mathcal{F}| = M$. If we use Hoeffding's inequality to derive a generalization bound based on n training examples, which of the following statements is true?

   A) The generalization bound is inversely proportional to the square root of $M \times n$.

   B) The generalization bound improves as the size of the hypothesis class M increases.

   C) The generalization bound worsens as the size of the hypothesis class M increases.

   D) The generalization bound is independent of the size of the hypothesis class M but dependent on n.

   Solution: *C.*
   Explanation: As M increases, the model has more flexibility to fit the training data, potentially leading to overfitting. Consequently, the generalization bound tends to worsen, indicating a higher risk of poor performance on unseen data.

4. Recall the following uniform deviation bound that you have seen in class,

$$p\Big(\exists f \in \mathcal{F},\ \Big|R(f) - \hat{R}(f, D_n)\Big| \ \geq\ \epsilon\Big) \leq 2|\mathcal{F}|\exp(-2 \cdot n \cdot \epsilon^2) \tag{1}$$

   Is the above bound applicable to the situation given here? If yes, then explain why and describe the $\mathcal{F}$ in this instance. And what is the value of $|\mathcal{F}|$?

   Solution: *Yes. We are looking for a good classifier among a set of 2^{kd} options. This is Hoeffding's inequality for finite hypothesis classes, so it can be applied.*

5. For some $\delta \in [0, 1]$ suppose we want that the following condition be true - that more than ϵ deviation of the true risk from the empirical risk is only at most $\delta-$probable for any $f \in \mathcal{F}$,

$$p\Big(\exists f \in \mathcal{F},\ \Big|R(f) - \hat{R}(f, D_n)\Big| \ \geq\ \epsilon\Big) \leq \delta \tag{2}$$

   What is a sufficient number of samples n needed to ensure the above? Answer in terms of k, d, ϵ and δ.

   Solution: *A sufficient number of samples n is that which solves the inequality $2|\mathcal{F}|\exp(-2 \cdot n \cdot \epsilon^2) \leq \delta$. This when solved for our situation with $|\mathcal{F}| = 2^{kd}$ gives the answer, $n \geq \frac{1}{2 \cdot \epsilon^2} \log\left(\frac{2^{1+kd}}{\delta}\right)$*

6. Can you conclude from all the given data and the analysis done so far that there is some linear function in this set $\mathcal{F}$ which has a population risk of zero?

   Solution: *No. For a start, no information has been given about the distribution of the data. So there is simply no information available to determine anything about the population risk of any member of the class $\mathcal{F}$.*

4 Bias and Variance

The material for this week is primarily a Python Notebook, available on Blackboard.

Please experiment with the code and get a ‘feeling’ for how the terms change as you (1) increase/decrease model complexity, (2) increase/decrease the amount of training data you provide to the model.

In addition, here are a few questiosn to test your knowledge....

1. What does high variance in a model indicate?

   A. Robust generalization
   B. High sensitivity to training data
   C. Balanced bias-variance trade-off
   D. Minimal model complexity

   B

2. What does the bias-variance decomposition help us understand?

   A. The parameter fitting process
   B. The trade-off between training and testing errors
   C. The trade-off between different sources of error in a predictive model
   D. The number of parameters in the model

   C

3. In the bias-variance decomposition, ‘bias’ refers to:

   A. The risk bias introduced by approximating real data with a model
   B. The risk due to fluctuations in the training data
   C. The risk due to the capacity of a model being too low
   D. The biased nature of the fitting process

   C

4. When a model is very simple (low complexity), what is a likely observation on bias and variance?

   A. Bias is high, variance is low
   B. Bias is low, variance is high
   C. Both bias and variance are high
   D. Both bias and variance are low

   A

5. Give a definition of the ‘double descent’ phenomenon, and its relevance to the question tackled by this module.

5 Ensemble Methods

1. What does the Ambiguity decomposition state?

   A. The loss of an ensemble is guaranteed to be either greater than or less than the average
   B. The classification error of an ensemble is guaranteed to be less than or equal to the average
   C. The risk of an ensemble is guaranteed to be less the average member risk
   D. The risk of an ensemble is ambiguous unless correctly specified by all members

   C

2. I have a dataset of images, where each image is either a cat, a dog, or a sheep. I ask you to help me train an ensemble of neural nets on this problem, such that I can exploit the error reduction guarantees of the Ambiguity decomposition. What is the output from the ensemble of models?

   A. A single label, stating 'cat', 'dog', or 'sheep'.
   B. A vector, taking the form $\bar{f}(\mathbf{x}) = Z^{-1} \prod_{i=1}^{m} f_i(\mathbf{x})^{1/m}$.
   C. A vector, taking the form $\bar{f}(\mathbf{x}) = \prod_{i=1}^{m} f_i(\mathbf{x})^{1/m}$.
   D. A vector of probabilities, given as the arithmetic mean of the networks

   B

3. You tell me to use the Bagging algorithm. I agree, and start coding, using a decision tree as my base learner. I find it doesn't quite work as well as I hope. The simplest option (i.e. with minimal code changes) to increase performance is:

   A. Edit my code to increase the number of samples, *with replacement*, and increase diversity
   B. Change the combiner to arithmetic mean: increasing diversity, but sacrificing individual error
   C. Edit the code to ensure each model corrects the mistakes of the previous model
   D. Edit my decision tree code, ensuring I pick sub-optimal features at each split, increasing diversity

   D

4. I have a neural network with 2 hidden nodes (i.e. low variance), and one with 500 hidden nodes (i.e. high variance). The dataset is a classification with 3 classes, and I combine my models via an arithmetic mean. I tell you I intend to apply the Random Forests algorithm, which is good for reducing bias. What's wrong with my reasoning? Give me at least 2 experimental setups that would be valid alternatives.

   Solution: *Come on, you can figure this out....*

5. In a Random Forest, what is the role of randomness at the split-points?

   A. It ensures that each tree in the forest learns on a different subset of data.
   B. It ensures that each tree in the forest learns on all features.
   C. It ensures that each tree in the forest learns from a different subset of features.
   D. It ensures that each tree in the forest learns from the correct subset of features.

   C

6. What is a bootstrap?

Solution: *A sample of n examples taken uniformly at random from the total n.*

7. In Boosting, what is the primary objective of subsequent models?

   A. To exploit errors made by the previous models, ensuring diversity
   B. To learn from the features that were previously ignored
   C. To increase the weight on correctly classified examples
   D. To decrease the weight on correctly classified examples
   E. To focus on errors made by the previous models

   $\boxed{\text{E}}$

6 Ensemble Theory

All questions assume the notation introduced in lecture notes/slides this week.

1. I am using the binary cross-entropy as a loss. I have 5 predictions of a class probability: $q_1 = 0.8$, $q_2 = 0.9$, $q_3 = 0.9$ $q_4 = 0.7$, and $q_5 = 0.95$. What is the centroid combiner prediction?

   Solution: *Take the NORMALIZED geometric mean of the values....* ≈ 0.87152341598.

2. I use the same predictions, but squared loss. What is the centroid combiner prediction?

   Solution: *Take the arithmetic mean of the values....* $(0.8 + 0.9 + 0.9 + 0.7 + 0.95)/5 = 0.85$.

3. Using squared loss on a problem, with an ensemble of size $M = 10$, I find my average bias is 0.2, and my average variance is 0.5. The overall loss of my ensemble model is 0.3. What is the diversity of my ensemble?

   Solution: $e = b + v - d$. *So,* $0.3 = 0.2 + 0.5 - d$. *Solve for d, gives* $d = 0.4$.
   The ensemble size being 10 is irrelevant to this calculation.

4. [**HARD ONE!**] The Ambiguity decomposition is proved in a very similar manner to the Bias/Variance decomposition. **Try to prove the Ambiguity decomposition for squared loss.** You can see the bias/variance proof in the notes.

   Solution: *Take the average loss...*

$$\frac{1}{M}\sum_i (f_i - y)^2 = \frac{1}{M}\sum_i (f_i - \bar{f} + \bar{f} - y)^2 \tag{3}$$

$$= \frac{1}{M}\sum_i \Big[(f_i - \bar{f})^2 + (\bar{f} - y)^2 + 2(f_i - \bar{f})(\bar{f} - y)\Big] \tag{4}$$

$$= \frac{1}{M}\sum_i \Big[(f_i - \bar{f})^2\Big] + (\bar{f} - y)^2 + \frac{1}{M}\sum_i \Big[2(f_i - \bar{f})(\bar{f} - y)\Big] \tag{5}$$

$$= \frac{1}{M}\sum_i \Big[(f_i - \bar{f})^2\Big] + (\bar{f} - y)^2 + 2(\bar{f} - y)\frac{1}{M}\sum_i \Big[(f_i - \bar{f})\Big] \tag{6}$$

$$= \frac{1}{M}\sum_i \Big[(f_i - \bar{f})^2\Big] + (\bar{f} - y)^2 + 2(\bar{f} - y)\Big[(\bar{f} - \bar{f})\Big] \tag{7}$$

$$= \frac{1}{M}\sum_i \Big[(f_i - \bar{f})^2\Big] + (\bar{f} - y)^2 \tag{8}$$

   Then simply rearrange terms to have the desired result.