1 Introduction

1. Write the formulas for the square loss, the zero-one loss, and the logarithmic loss.

    - Square loss: $\ell(y, \hat{y}) = (y - \hat{y})^2$
    - Zero-one loss: $\ell(y, \hat{y}) = \begin{cases} 0 & y = \hat{y} \\ 1 & y \neq \hat{y} \end{cases}$
    - Logarithmic loss: $\ell(y, \hat{y}) = \begin{cases} \log \frac{1}{\hat{y}} & y = 1 \\ \log \frac{1}{1-\hat{y}} & y = 0 \end{cases}$

2. What does a learning algorithm receive in input? And what does it produce in output?

    A learning algorithm A receives in input a training set S, that is a subset of examples taked from the set of all data points. A produces in output a predictor $f : \mathcal{X} \to \mathcal{Y}$.

3. Write the mathematical formula defining the training error of a predictor h.

$$\ell_S(f) = \frac{1}{m} \sum_{t=1}^{m} \ell(y_t, f(x_t))$$

$$m = |S|$$

4. Write the mathematical formula defining the ERM algorithm over a class H of predictors. Define the main quantities occurring in the formula.

$$A(S) \in \operatorname*{argmin}_{f \in \mathcal{F}} \ell_S(f)$$

    - A: learning algorithm
    - S: training set
    - $\mathcal{F}$: set of all possible predictors

5. Explain in words how overfitting and underfitting are defined in terms of behavior of an algorithm on training and test set.

    When the training error is low but the test error is high, we have overfitting, that is, the predictor has learned also the noise from the training set. On the other side, when the training error is high we have underfitting.

6. Name and describe three reasons why labels may be noisy. The labels can be noisy due to:

    (a) Human uncertainty: if the labels are assigned by humans, it may happen that two different person could have different opinion and so different labels at the same data point;

    (b) Epistemic uncertainty: the data points are represented with a vector that not contains enough information for uniquely determine the label;

    (c) Aleatoric uncertainty: the data points are obtained through some noisy measurement.

2 NN

7. Is k-NN more likely to overfit when k is large or small?

    When is smaller. More smaller is k, more the k-NN predictor make the prediction more influenced by the training set, capturing also the noise.

3 Tree predictors

8. Write a short pseudo-code for building a tree classifier based on a training set S.

   (a) Initialization:

   - Create the T tree with only one leaf ℓ
   - $S_\ell = S$
   - $y_\ell =$ most frequent label in S_ℓ

   (b) Loop:

   - Pick a leaf ℓ and replace it with a node v and create two children leaves ℓ' and ℓ''
   - Pick a feature i and a test $f : \mathcal{X}_i \to \{1, 2\}$
   - Associate the test f with the node v
   - $S_{\ell'} = \{(\boldsymbol{x_t}, y_t) \in S_\ell : f(x_{t,i}) = 1\}$
   - $S_{\ell''} = \{(\boldsymbol{x_t}, y_t) \in S_\ell : f(x_{t,i}) = 2\}$
   - Associate to ℓ' the most frequent label in $S_{\ell'}$
   - Associate to ℓ'' the most frequent label in $S_{\ell''}$

9. What is the property of a splitting criterion ψ ensuring that the training error of a tree classifier does not increase after a split?

   For ensuring that the training error does not increase, ψ must be a concave function.

10. Write the formulas for at least two splitting criteria ψ used in practice to build tree classifiers.

    (a) $\psi_1(\alpha) = \min\{\alpha, 1 - \alpha\}$

    (b) $\psi_2(p) = 2p(1 - p)$

    (c) $\psi_3(p) = \sqrt{p(1 - p)}$

4 Statistical risk

11. Write the formula for the statistical risk of a predictor h with respect to a generic loss function and data distribution.

$$\ell_{\mathcal{D}}(h) = \mathbb{E}[\ell(Y, h(X))]$$

12. Write the formula for the Bayes optimal predictor for a generic loss function and data distribution.

$$f^*(\boldsymbol{x}) = \underset{\hat{y} \in \mathcal{Y}}{\operatorname{argmin}} \, \mathbb{E}[\ell(Y, \hat{y}) | \boldsymbol{X} = \boldsymbol{x}]$$

13. Write the formula for Bayes optimal predictor and Bayes risk for the zero-one loss.

$$f^*(\boldsymbol{x}) = \begin{cases} +1 & \eta(\boldsymbol{x}) \geq \frac{1}{2} \\ -1 & \eta(\boldsymbol{x}) < \frac{1}{2} \end{cases} \qquad \eta(x) = \mathbb{P}(Y = 1 | \boldsymbol{X = x})$$

$$\ell_{\mathcal{D}}(f^*) = \mathbb{E}[\min\{\eta(\boldsymbol{X}), 1 - \eta(\boldsymbol{X})\}]$$

14. Can the Bayes risk for the zero-one loss be zero? If yes, then explain how.

    If $\ell_{\mathcal{D}}(f^*) = 0$ it means that the predictor f^* never errs. This can only happen if:

    (a) The data are perfectly separable

    (b) The distribution $\mathcal{D}$ is known

15. Write the formula for Bayes optimal predictor and Bayes risk for the square loss.

$$f^*(x) = \mathrm{Var}[Y|X = x] \qquad \ell_{\mathcal{D}}(f^*) = \mathbb{E}[\mathrm{Var}[Y|X]]$$

16. Explain in mathematical terms the relationship between test error and statistical risk.

$\ell_{\mathcal{D}}(h)$ cannot be directly calculeted because the distribution $\mathcal{D}$ is unknown. We thus have to estimate it. We assume that the test set $S' = \{(x'_1, y'_1), \dots, (x'_n, y'_n)\}$ is generated from a random draw from $\mathcal{D}$ and so $(X'_t, Y'_t) \sim \mathcal{D}$ for every $t = 1, \dots, n$. Therefore:

$$\mathbb{E}[\ell(Y'_t, h(X'_t))] = \ell_{\mathcal{D}}(h)$$

For estimating this value we use the sample mean of the loss function over S', that is the test error $\ell_{S'}(h)$. Is important that h doesn't have to depend on S'.

17. State the Chernoff-Hoeffding bounds.

Let $Z_1, \dots, Z_n$ be i.i.d. such that $\forall i \; Z_i \in [0, 1]$. For any $\varepsilon > 0$:

$$\mathbb{P}\left(\frac{1}{n}\sum_{t=1}^{n} Z_t > \mu + \varepsilon\right) \le e^{-2\varepsilon^2 n} \quad \wedge \quad \mathbb{P}\left(\frac{1}{n}\sum_{t=1}^{n} Z_t < \mu - \varepsilon\right) \le e^{-2\varepsilon^2 n}$$

18. Write the bias-variance decomposition for a generic learning algorithm A and associate the resulting components to overfitting and underfitting.

$$\begin{aligned} \ell_{\mathcal{D}}(h_S) &= \ell_{\mathcal{D}}(h_S) - \ell_{\mathcal{D}}(h^*) + && \text{(estimation error)} \\ &= \ell_{\mathcal{D}}(h^*) - \ell_{\mathcal{D}}(f^*) + && \text{(bias error)} \\ &= \ell_{\mathcal{D}}(f^*) && \text{(Bayes risk)} \end{aligned}$$

If the estimation error dominates the bias error we have overfitting while in the opposite case we have underfitting.

19. Write the upper bound on the estimation error of ERM run on a finite class $\mathcal{H}$ of predictors.

$$\mathbb{P}\left(\ell_{\mathcal{D}}(h_S) - \ell_{\mathcal{D}}(h^*) \le \sqrt{\frac{2}{m} \ln \frac{2|\mathcal{H}|}{\delta}}\right) \ge 1 - \delta$$

$$\delta = 2|\mathcal{H}|e^{-m\varepsilon^2/2}$$

5 Risk analysis for tree predictor

20. What is the number of nodes that a tree classifier must have so that by assigning labels to its leaves we can compute any function of the form $h : \{0, 1\}^d \to \{-1, 1\}$?

For each possible $x \in \{0, 1\}^d$ we need a leaf with the corresponding label. So the three has 2^d leaves. Because the tree is complete and has 2^d leaves, it has $2^{d+1} - 1$ nodes.

21. Write an upper bound on the set $\mathcal{H}_N$ of all classifiers computed by complete binary tree predictors with exactly N nodes on d binary features.

$$|\mathcal{H}_N| \le (2de)^N$$

22. Write the upper bound on the estimation error of ERM run on a the class of complete binary tree predictors with at most N nodes on d binary features.

$$\ell_{\mathcal{D}}(h_S) \le \ell_{\mathcal{D}}(h^*) + \sqrt{\frac{2}{m}\left(\ln\frac{2}{\delta} + N\left(\ln 2d + 1\right)\right)}$$

23. How many bits are sufficient to encode an arbitrary tree predictor with N nodes on d binary features so that no tree has an encoding that is a prefix of the encoding of a different tree?

$$(N+1)\lceil\log_2(d+3)\rceil + 2\lfloor\log_2 N\rfloor + 1$$

24. Write the bound on the difference between risk and training error for an arbitrary complete binary tree classifier h on d binary features in terms of its number N_h of nodes. Bonus points if you provide a short explanation on how this bound is obtained.

$$\ell_{\mathcal{D}}(h) \le \ell_S(h) + \sqrt{\frac{1}{2m}\left(|\sigma(h)| + \ln\frac{2}{\delta}\right)}$$

$$|\sigma(h)| = O(N_h \log d)$$

6 Hyperparameter tuning and risk estimation

25. Write the formula for the K-fold cross validation estimate. Explain the main quantities occurring in the formula.

$$\ell_{S_i}(h_i) = \frac{K}{m}\sum_{(\boldsymbol{x},y)\in S_i} \ell(y, h_i(\boldsymbol{x})) \qquad \text{(mean error of every fold)}$$

$$\ell_S^{\text{cv}}(A) = \frac{1}{K}\sum_{i=1}^{K} \ell_{S_i}(h_i) \qquad \text{(average error of all folds)}$$

$$K = \text{number of partitions} \qquad m = |S_i|$$

26. Write the pseudo-code for computing the nested cross validation estimate.

```
$S = S_1 \cup \cdots \cup S_K$
for $i = 1, \dots, K$ do
    $S_{-i} = S \setminus S_i$
    Run CV on $S_{-i}$ for each $\theta \in \Theta_0$ and find $\theta_i = \operatorname*{argmin}_{\theta\in\Theta_0} \ell^{\text{cv}}_{S_{-i}}(A_\theta)$
    $h_i = A_{\theta_i}(S_{-i})$
    $\varepsilon_i = \ell_{S_i}(h_i)$
end
Output: $(\varepsilon_1 + \cdots + \varepsilon_K)/K$
```

7 Consistency and nonparametric algorithms

27. Write the mathematical definition of consistency for an algorithm A.

An algorithm A is consistent if, for any data distribution $\mathcal{D}$:

$$\lim_{m\to\infty} \mathbb{E}[\ell_{\mathcal{D}}(A(S_m))] = \ell_D(f^*)$$

28. Write the statement of the no-free-lunch theorem.

For any sequence $a_1, a_2, \dots$ of positive numbers such that $\frac{1}{16} \ge a_1 \ge a_2 \ge \cdots > 0$ and for any consistent algorithm A for binary classification with zero-one loss, there exists a data distribution $\mathcal{D}$ such that:

$$\ell_{\mathcal{D}}(f^*) = 0 \qquad \mathbb{E}[\ell_{\mathcal{D}}(A(S_m))] \ge a_m \qquad \forall m \ge 1$$

29. Write the mathematical definition of nonparametric learning algorithm. Define the main quantities occurring in the formula.

    An algorithm A is nonparametric if:

    $$\lim_{m \to \infty} \min_{h \in \mathcal{H}_m} \ell_{\mathcal{D}}(h) = \ell_{\mathcal{D}}(f^*)$$

    - $\mathcal{H}_m$: set of all predictors generated by any training set of size m

30. Name one nonparametric learning algorithm and one parametric learning algorithm.

    - Nonparametric: k-NN
    - Parametric: linear regression

31. Write the mathematical conditions on k ensuring consistency for the k-NN algorithm. Provide an example of a choice of k that makes k-NN consistent as a function of the training set size m.

    If k is a function k_m of the training set size, k_m-NN is consistent if:

    $$k_m \to \infty \quad \wedge \quad k_m = o(m)$$

    An example is $k_m = \log m$.

32. Write the formula for the Lipschitz condition in a binary classification problem. Define the main quantities occurring in the formula.

    In a distribution, the function $\eta(\boldsymbol{x}) = \mathbb{P}(Y = 1|\boldsymbol{X} = \boldsymbol{x})$ is a Lipschitz function on $\boldsymbol{X}$ if:

    $$\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X} \quad \exists c : 0 < c < \infty \quad |\eta(\boldsymbol{x}) - \eta(\boldsymbol{x}')| \leq c||\boldsymbol{x} - \boldsymbol{x}'||$$

33. Write the rate at which the risk of a consistent learning algorithm for binary classification vanishes as a function of the training set size m and the dimension d under Lipschitz assumptions.

    Assuming Lipschitz, the rate of convergence is $m^{\frac{-1}{d+1}}$.

34. Explain the curse of dimensionality.

    The convergence rate of $m^{\frac{-1}{d+1}}$ in the nonparametric case implies that to get ε-close to Bayes risk, we need a training set size m of order of $\varepsilon^{-(d+1)}$. This show that if d is high we need an exponentially high number of training set elements. This is known as curse of dimensionality.

8 Risk analysis for NN

35. Write the bound on the risk of the 1-NN binary classifier under Lipschitz assumptions.

    $$\begin{aligned}\mathbb{E}[\ell_{\mathcal{D}}(\text{1-NN}(S_M))] \leq& 2\ell_{\mathcal{D}}(f^*) + c\mathbb{E}\left[||\boldsymbol{X} - \boldsymbol{X}_{\pi(S,\boldsymbol{X})}||\right] \\ \leq& 2\ell_{\mathcal{D}}(f^*) + 4c\sqrt{d}m^{\frac{-1}{d+1}}\end{aligned}$$

    $$\pi(S, \boldsymbol{x}) = \operatorname*{argmin}_{t=1,\dots,m} ||\boldsymbol{x} - \boldsymbol{x}_t||$$

    For $m \to \infty$ we have $\ell_{\mathcal{D}}(h_S) \leq \mathbb{E}[\ell_{\mathcal{D}}(h_S)] \leq 2\ell_{\mathcal{D}}(h_S)$.

9 Linear predictor

36. Can the ERM over linear classifiers be computed efficiently? Can it be approximated efficiently? Motivate your answers.

    ERM over linear classifier is:

    $$h_S = \operatorname*{argmin}_{w \in \mathbb{R}:||w||=1} \frac{1}{m} \sum_{t=1}^{m} \mathbb{I}\{y_t w^\top x_t \leq 0\}$$

    It is unlikely to find an efficient implementation of ERM because the problem of finding w such that $y_t w^\top x_t \leq 0$ for at most k indices is NP-complete. So, cosidering that almost certainly $P \neq NP$, there are no algorithms that solve the problem in polynomial time.

    The only way to compute efficiently a linear classifier is with linearly separable data.

37. Write the system of linear inequalities stating the condition of linear separability for a training set in binary classification.

    A training set is linearly separable if the following system has at least one solution:

    $$\begin{cases} y_1 w^\top x_1 > 0 \\ \dots \\ y_m w^\top x_m > 0 \end{cases}$$

38. Write the pseudo-code for the Perceptron algorithm.

```
Data: $S = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_m, y_m)\}$
$w \leftarrow (0, \dots, 0)$
while true do
    for $t = 1, \dots, m$ do
        if $y_t w^\top \boldsymbol{x}_t \leq 0$ then
            $w \leftarrow w + y_t \boldsymbol{x}_t$
    end
    if no update then break
end
Output: $w$
```

39. Write the statement of the Perceptron convergence theorem.

    Let S linearly separable; The Perceptron always terminates after a number of updates not bigger than:

    $$\left(\min_{u:\gamma(u) \geq 1} ||u||^2 \right) \left(\max_{t=1,\dots,m} ||x_t||^2 \right)$$

    $$\gamma(u) = \min_{t=1,\dots,m} y_t u^\top x_t$$

40. Write the closed-form formula (i.e., not the argmin definition) for the Ridge Regression predictor. Define the main quantities occurring in the formula.

    $$w_{S,\alpha} = (\alpha I + S^\top S)^{-1} S^\top y$$

    - $S^\top = [x_1, \dots, x_m]$
    - $\alpha > 0$

10 Online Gradient Descent

41. Write the pseudo-code for the projected online gradient descent algorithm.

```
Parameters: $\eta > 0,\ U > 0$
$w_1 = 0$
for $t = 1, 2, \ldots$ do
    $w'_{t+1} = w_t - \frac{\eta}{\sqrt{t}} \nabla \ell_t(w_t)$
    $w_{t+1} = \underset{w:||w|| \le U}{\text{argmin}}\ ||w - w'_{t+1}||$
end
```

42. Write the upper bound on the regret of projected online gradient descent on convex functions. Define the main quantities occurring in the bound.

$$\underbrace{\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{w}_t)}_{\ell_T(\boldsymbol{w})} \le \underbrace{\min_{\boldsymbol{u}:||\boldsymbol{u}||\le U}\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{u})}_{\ell_T(u^*)} + \underbrace{UG\sqrt{\frac{8}{T}}}_{\text{regret}}$$

- $\ell_T(\boldsymbol{w})$: average loss of the algorithm up to step T
- $\ell_T(\boldsymbol{u}^*)$: average loss of the best finded predictor up to step T
- U: maximum radius of the sphere where the projection is performed
- G: bound on the norm of the gradient of the loss function

43. Write the upper bound on the regret of online gradient descent on σ-strongly convex functions. Define the main quantities occurring in the bound.

$$\underbrace{\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{w}_t)}_{\ell_T(\boldsymbol{w})} \le \underbrace{\min_{\boldsymbol{u}\in\mathbb{R}^d}\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{u})}_{\ell_T(u^*)} + \underbrace{\frac{G^2(1+\ln T)}{2\sigma T}}_{\text{regret}}$$

- $\ell_T(\boldsymbol{w})$: average loss of the algorithm up to step T
- $\ell_T(\boldsymbol{u}^*)$: average loss of the best finded predictor up to step T
- G: bound on the norm of the gradient of the loss function

44. Write the formula of the hinge loss.

$$h_t(\boldsymbol{u}) = \max\{0, 1 - y_t \boldsymbol{u}^\top \boldsymbol{x}_t\}$$

45. Write the mistake bound for the Perceptron run on an arbitrary data stream for binary classification. Define the main quantities occurring in the bound.

$$\forall \boldsymbol{u} \in \mathbb{R}^d \quad M \le \sum_{t=1}^{T} h_t(\boldsymbol{u}) + (||\boldsymbol{u}||X)^2 + ||\boldsymbol{u}||X\sqrt{\sum_{t=1}^{T} h_t(\boldsymbol{u})}$$

- M: number of mistakes made by the Perceptron
- $h_t(\boldsymbol{u}) = \max\{0, 1 - y_t \boldsymbol{u}^\top x_t\}$ (hinge loss)
- $X = \max_t ||x_t||$

11 Kernel functions

46. Write the formula for the polynomial kernel of degree n.

$$K_n(\boldsymbol{x}, \boldsymbol{x}') = (1 + \boldsymbol{x}^\top \boldsymbol{x}')^n$$

47. Write the formula for the Gaussian kernel with parameter γ.

$$K_n(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{1}{2\gamma}||\boldsymbol{x} - \boldsymbol{x}'||^2\right)$$

$$\gamma > 0$$

48. Write the pseudo-code for the kernel Perceptron algorithm.

```
$S \leftarrow \{\}$;
for $t = 1, 2, \dots$ do
    Get the next example $(\boldsymbol{x}_t, y_t)$;
    $\hat{y}_t = \operatorname{sgn}\left(\sum_{s \in S} y_s K(\boldsymbol{x}_s, \boldsymbol{x}_t)\right)$;
    if $\hat{y}_t \neq y_t$ then
        $S \leftarrow S \cup \{t\}$
    end
end
```

49. Write the mathematical definition of the linear space $\mathcal{H}_K$ of functions induced by a kernel K.

$$\mathcal{H}_K = \left\{\sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}_i, \cdot) : \boldsymbol{x}_1, \dots, \boldsymbol{x}_N \in \mathcal{X}, \alpha_1, \dots, \alpha_N \in \mathbb{R}, N \in \mathbb{N}\right\}$$

50. Let f be an arbitrary element of the linear space $\mathcal{H}_K$ induced by a kernel K. Write $f(x)$ in terms of K.

$$f(x) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x})$$

51. Write the mistake bound of the Perceptron convergence theorem when the Perceptron is run with a kernel K. Define the main quantities occurring in the bound.

$$M \leq \left(\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)\left(\max_t\{K(\boldsymbol{x}_t, \boldsymbol{x}_t)\}\right)$$

52. Write the closed-form formula (i.e., not the argmin definition) of the kernel version of the Ridge Regression predictor.

$$w^\top \boldsymbol{x} = \boldsymbol{y}^\top(\alpha I + \boldsymbol{K})^{-1} k(\boldsymbol{x})$$

- $\boldsymbol{K}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- $k(\boldsymbol{x}) = (K(\boldsymbol{x}_1, \boldsymbol{x}), \dots, K(\boldsymbol{x}_m, \boldsymbol{x}))$

12 Support Vector Machine

53. Write the convex optimization problem with linear constraints that defines the SVM hyperplane in the linearly separable case.

$$\min_{w \in \mathbb{R}^d} \frac{1}{2}||w||^2$$

$$s.t. \;\; y_t w^\top x_t \leq 1 \quad t = 1, \dots, m$$

54. Write the unconstrained optimization problem whose solution defines the SVM hyperplane when the training set is not necessarily linearly separable.

$$\min_{w \in \mathbb{R}^d} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{t=1}^{m} h_t(w)$$

$$h_t(w) = \max\{0, 1 - y_t w^\top x_t\}$$

55. Write the update rule of Pegasos.

$$w_{t+1} = w_t - \eta_t \nabla \ell_{Z_t}(w_t)$$

56. Write the bound on the expected value of the SVM objective function achieved by Pegasos. Provide also a bound on the expected squared norm of the loss gradient in terms of a bound X on the norm of the training points.

$$\mathbb{E}[F(\bar{w})] \leq F(w^*) + \frac{\mathbb{E}[G^2]}{2\lambda T}(\ln T + 1)$$

$$F(w) = \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{t=1}^{m} h_t(w) \quad , \quad \bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t \quad , \quad w^* = \underset{w \in \mathbb{R}^d}{\text{argmin}}(F(w))$$

$$G = \max_{t=1,\dots,T} ||\ell_{S_t}(w_t)|| \quad , \quad \lambda > 0$$

$$\mathbb{E}[F(\bar{w})] \leq F(w^*) + \frac{2X^2}{\lambda T}(\ln T + 1)$$

13 Stability and risk control for SVM

57. Write the definition of ε-stability for a learning algorithm.

An algorithm is ε-stable if:

$$\mathbb{E}[\ell(h_{S^{(t)}}, \boldsymbol{Z}_t) - \ell(h_S, \boldsymbol{Z}_t)] \leq \varepsilon$$

$$\boldsymbol{Z}_t = (\boldsymbol{x}_t, y_t)$$

58. Write the value of ε for which SVM is known to be stable. The value depends on the radius X of the ball where the training datapoints live, the training set size m, and the regularization coefficient λ.

$$\varepsilon = \frac{4X^2}{\lambda m}$$

59. Write the mathematical conditions on the regularization coefficient λ ensuring consistency for the SVM algorithm wih Gaussian kernel.

Let λ be a function λ_m of the size of S. SVM is consistent with Gaussian Kernel if:

$$\lim_{m \to \infty} \lambda_m = o(1) \ \wedge \ \lim_{m \to \infty} \lambda_m = w\left(m^{-\frac{1}{2}}\right)$$

14 Neural Networks and Deep Learning

60. Consider the class $\mathcal{F}_d$ of all functions of the form $f : \{1, 1\}^d \to \{1, 1\}$. Let $\mathcal{F}_{G,\text{sgn}}$ be the class of functions computed by a feedforward neural networks with the sgn activation function and graph $G = (V, E)$. Provide asymptotic upper and lower bounds on $|V|$ such that $\mathcal{F}_d \subseteq \mathcal{F}_{G,\text{sgn}}$.
$$|V| = \Omega(2^{d/3})$$

61. Define a class of neural networks for which the ERM problem with the square loss is probably NP-hard.

    For any $k \in \mathbb{N} : k \geq 2$, the problem:
$$\min_{f \in F_{k,\sigma}} \ell_S(f) \text{ such that } \ell_S(f) = \frac{1}{m}\sum_{t=1}^{m}(f(\boldsymbol{x}) - y_t)^2$$
    is NP-HARD.

62. Write the update line of the stochastic gradient descent algorithm. Explain the main quantities.
$$w_{i,j} \leftarrow w_{i,j} - \eta_t \frac{\partial \ell_{Z_t}(w)}{\partial w_{i,j}} \qquad (i, j) \in E$$
    - $w_{i,j}$ is the weight of the edge $(i, j) \in E$
    - η_t is the learning rate at the iteration t
    - $\ell_{\boldsymbol{Z}_t}(\boldsymbol{w})$ is the loss function $\ell(y_t, \hat{y}_t)$ where Z_t is the index of a random training example.

15 Logistic Regression and Surrogate Loss Function

63. Write the definition of logistic loss for logistic regression with linear models.
$$\ell_t(w) = \log_2\left(1 + e^{-y_t \boldsymbol{w}^\top \boldsymbol{x}_t}\right)$$

64. Define the surrogate losses used by SVM and AdaBoost.
    - SVM: $\ell(y, \hat{y}) = \max\{0, 1 - y_t\hat{y}\}$
    - AdaBoost: $\ell(y, \hat{y}) = e^{-y\hat{y}}$

65. Write the definition of consistency for surrogate losses.

    A surrogate loss function ℓ is consistent if:
$$\forall x \in \mathcal{X} \quad \text{sgn}(g^*) = f^*$$
$$g^*(x) = \operatorname*{argmin}_{\hat{y} \in \mathbb{R}} \mathbb{E}[\ell(\hat{y}, Y)|X = x]$$

66. Write a sufficient condition for consistency of a surrogate loss.

    If a surrogate loss function ℓ is such that:
$$\forall y \in \{-1, 1\} \quad \ell'(y, 0) \text{ exists } \quad \wedge \quad \ell'(y, 0) < 0$$
    then ℓ is consistent.

67. Write the formula for Bayes optimal predictor and Bayes risk for the logistic loss.
$$g^*(x) = \ln\left(\frac{\eta(x)}{1 - \eta(x)}\right) \qquad , \qquad \ell_D(g^*) = H(Y|\boldsymbol{X})$$
$$H(Y|\boldsymbol{X}) : \text{conditional entropy}$$