Exercises

1	2	3	4	5	6	7	8	9	10
11	12	13							

Surname, First name

____

1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9
0	0	0	0	0	0	0	0

Machine Learning 1 (52041MAL6Y)
ML1 Main Exam 2022

_______________________*Resit for Machine Learning 1 - 25 October 2022*_________________________

Pay attention to the following

- **Write your name and student number on the front page** (don't forget to mark the digits)
- Write all your answers on this exam booklet; it will be scanned and digitally graded.
- Write your **answers inside the boxes** (it is ok, to slightly go over the margins though).
- The answer boxes should be large enough for your answer
- **If you need to empty the answer box in order to start over, ask the invigilator for a blank sticker**

During the exam you are allowed to use:

- One double-sided handwritten cheat sheet.
- A calculator (though not strictly necessary)

About the multiple choice questions:

- You should fill the boxes and not just check them. E.g.

  Correct marking: ■ ● Incorrect marking: ☒ ☑ ⊗ ⓐ

- In case you want to correct your answer, clear indicate this (e.g. by filling all boxes and use another indicator such as an arrow to indicate your choice). We can then resolve this during grading.
- The grading of the multiple choice questions is based partial grading; not giving all correct options will give you some points, but not all. Marking incorrect options whilst also marking correct options also reduces the nr of points.

About the open questions:

- We work with a partial grading: you can get points even if you don't manage to solve the full question.

Please scan over the questions before you start to get an impression of the content

- In total **40 points** can be earned (+ 3 bonus) divided over the following 5 categories.
- You have **3 hours** for the exam (except for those with pre-approved extensions)
- Exercises 1-9: Multiple choice (9.0 pts)
- Exercise 10: Gamma distribution (5.5 pts + 2.0 pts bonus)
- Exercise 11: Pet Detective (6.0 pts)
- Exercise 12: Modeling Muons (8.0 pts)
- Exercise 14: Polar Coordinate SVM (11.5 pts + 1.0 pt bonus)

MC: Evil Likelihood Model

1p **1** Suppose you wish to devise an "evil likelihood model", which will make your model likelihood $p(\mathbf{x}|\mathbf{w})$ as low as possible. In this case, we wish to find the minimum likelihood solution for the model parameters $\mathbf{w}$. Which statements are true?

□ In the case of a linear regression, we can find the unique solution to this problem.

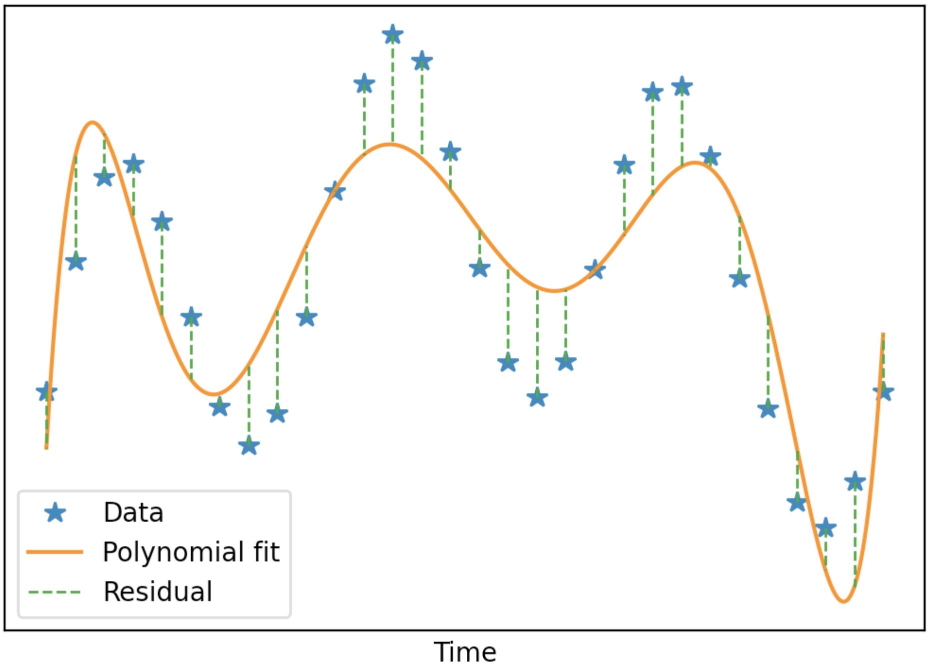
□ In the case of linear regression, the minimum likelihood estimate is equal to the maximum likelihood solution, but with opposite signs.

□ The goal of such model is to maximize the function: $-\log p(\mathbf{x}|\mathbf{w})$.

□ The goal of such model is to maximize the function: $\frac{1}{p(\mathbf{x}|\mathbf{w})}$.

MC: FLAC compression

1p



2

Audio is stored as sequence of measurements of the waveform at discrete time intervals. FLAC is a way of compressing audio by least-squares fitting a polynomial and storing the residual. Weights are stored at full precision, while the residual is compressed further. Which statements are true?

□ The audio data is i.i.d.

□ The weights for the polynomial fit have a closed-form solution.

□ Increasing the order M of the polynomial causes the residual to shrink

□ Adding L2 regularisation can improve compression performance.

MC: I.i.d., Conditional Independence and GPs

1p **3** Let $\mathcal{D} = \{(x_i, t_i)\}_{i=1}^N$ with samples given by $t_i = f(x_i)$ with $f \sim GP(m(\cdot), k(\cdot, \cdot))$ a random function according to a Gaussian process with mean function m and kernel k. Which statements are true?

□ t_i is not a random variable.

□ $t_i \sim p(t)$ is i.i.d. relative to some $p(t)$.

□ $t_i \sim p(t|x_i)$ is i.i.d. relative to some $p(t|x)$

□ $\mathrm{Cov}[t_i, t_j] = 0$ for any $i \neq j$

□ $\mathrm{Cov}[t_i, t_j] = k(t_i, t_j)$

□ $\mathrm{Cov}[t_i, t_j] = k(x_i, x_j)$

MC: Valid Kernels

1p **4** Which of the following kernels are valid? Let $x, x' \in \mathbb{R}^d$ be two vectors of the same dimensionality d.

□ $k(x, x') = 1$

□ $k(x, x') = x \cdot x'$

□ $k(x, x') = 1 + x \cdot x'$

□ $k(x, x') = \min(x, x')$, for $x, x' \in \mathbb{R}$

□ $k(x, x') = \exp(x + x')$, for $x, x' \in \mathbb{R}$

□ $k(x, x') = (1 + x \cdot x')^2$

MC: Overfitting and model complexity

1p **5** Which of the following statements are true? Check all that apply

□ Higher complexity models are more prone to overfitting and typically have lower variance

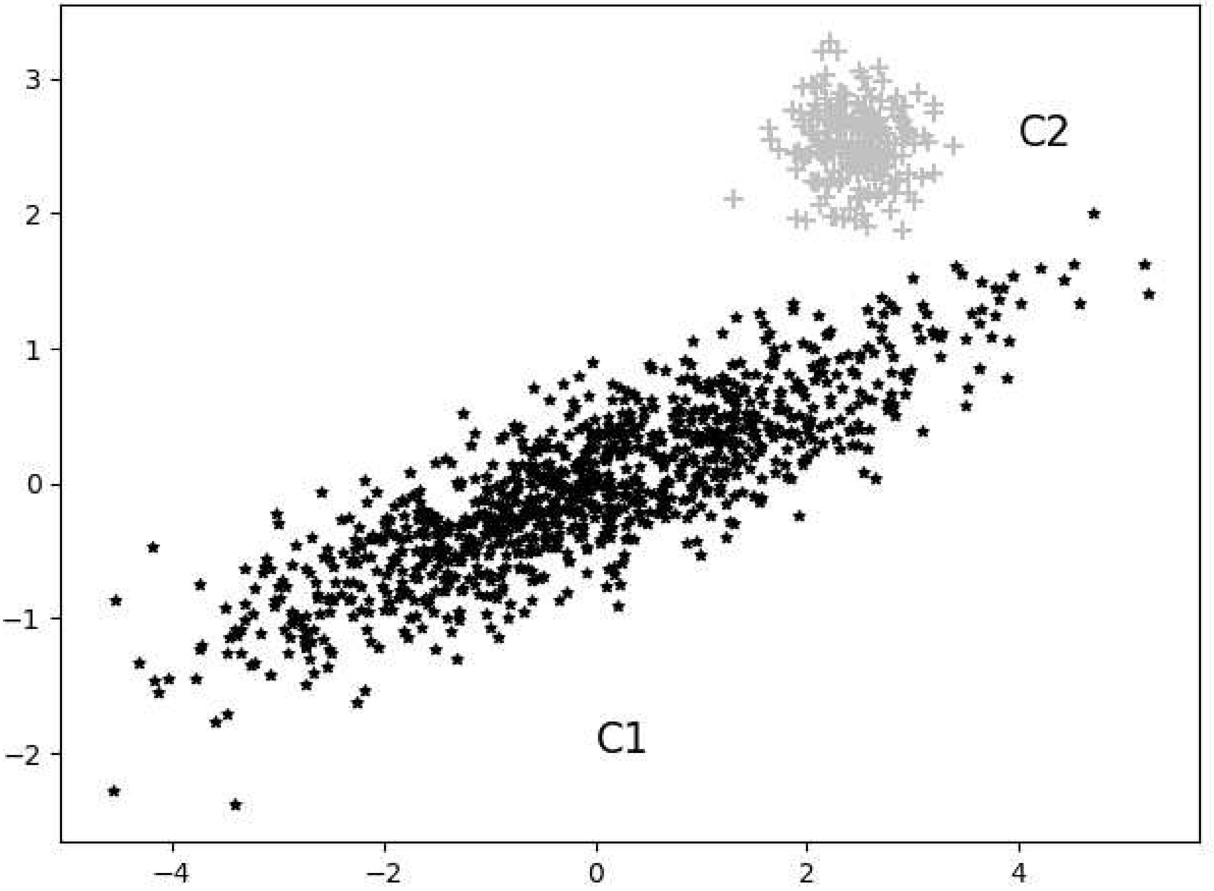
□ Only adding more data for training a learner with high bias may not reduce the test error.

□ Overfitting may arise when relevant features are missing in the data

□ Increasing the depth of a neural network will always reduce the test error.

MC: Two clusters

1p **6** We have the following dataset, where the samples belong to two different classes $C1, C2$. Which of the following statements are true? (*Note: I_2 denotes the 2x2 identity*



matrix.)

□ The conditional distribution for $C1$ can be accurately modeled with a Gaussian $\mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1)$ for some mean vector $\boldsymbol{\mu}_1 \in \mathbb{R}^2$ and some covariance matrix Σ_1.

□ A Gaussian Mixture Model with 2 Gaussian components may fit the data well.

□ K-means with 2 means may fit the data well.

□ The conditional distribution for $C2$ can be accurately modeled with a Gaussian $\mathcal{N}(\boldsymbol{\mu}_2, \Sigma_2)$ for some mean vector $\boldsymbol{\mu}_2 \in \mathbb{R}^2$ and some covariance matrix Σ_2.

□ The conditional distribution for $C2$ can be accurately modeled with a Gaussian $\mathcal{N}(\boldsymbol{\mu}_2, s \cdot I_2)$ for some mean vector $\boldsymbol{\mu}_2 \in \mathbb{R}^2$ and some scalar $s > 0$.

□ The conditional distribution for $C1$ can be accurately modeled with a Gaussian $\mathcal{N}(\boldsymbol{\mu}_1, s \cdot I_2)$ for some mean vector $\boldsymbol{\mu}_1 \in \mathbb{R}^2$ and some scalar $s > 0$.

MC: Probabilistic models

1p **7** In classification we consider three models: discriminant functions, probabilistic generative models and probabilistic discriminative models. The following are statements about probabilistic generative models and probabilistic discriminative models. Which are true?

□ Logistic regression is a probabilistic discriminative model.

□ In probabilistic discriminative models, the prior probability of class $p(C)$ is modeled.

□ In generative models, the class conditional probability $p(\mathbf{x}|C)$ are modeled.

□ In probabilistic discriminative models, the posterior probabilities $p(C|\mathbf{x})$ are modeled directly.

MC: SVM

1p **8** Consider the following SVM optimization problem. Which statements are true?

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}||\mathbf{w}||^2 + \frac{1}{\lambda}\sum_{n=1}^{N} \xi\,, \quad \text{s.t.} \begin{cases} \forall_n : t_n(\mathbf{w}^\top \mathbf{x}_n + b) & \geq 1 - \xi \\ \forall_n : \xi & \geq 0 \end{cases}$$

□ For large λ we expect a more complex decision boundary than for small λ.

□ For large λ we expect more support vectors than for small λ.

□ For large λ we expect a less complex decision boundary than for small λ.

□ For large λ we expect less support vectors than for small λ.

MC: Neural Networks

1p **9** Consider a neural network with $L = 5$ layers. Let us denote $w_{ij}^{(l)}$ the weights in each layer, the number of features (the width) in each hidden layer with M, the used hidden activation functions with h, and the error of the model with respect to the n^{th} data point with E_n. Which statements are true?

□ Updating the weights in layer $l = 2$ requires to perform the forward pass only up to layer 2.

□ Updating the weights in layer $l = 3$ requires computation of the backward pass down to all layers.

□ In order for back-propagation to work the network is not allowed to contain skip connections.

□ Optimizing a neural network with stochastic gradient descent means updating the weights via $w_{ij}^{(l)} = w_{ij}^{(l)} - \eta \frac{\partial E_n}{\partial w_{ij}^{(l)}}$, with η a hyperparameter.

□ Using $h(a) = a^2$, the neural network can in theory represent any function $\phi(x)$ up to arbitrary precision by scaling up M.

Gamma Distribution

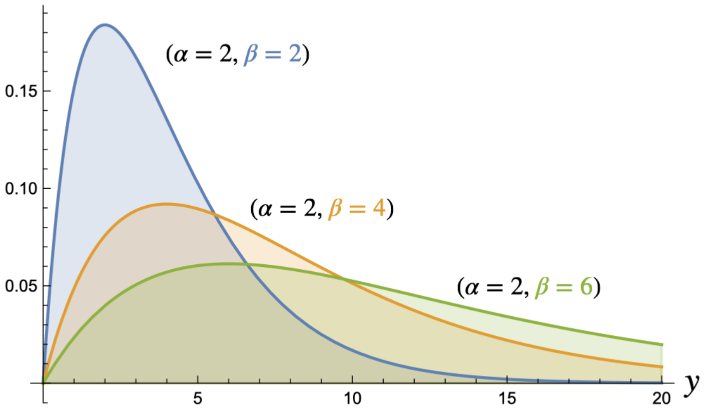
In linear regression we assume that $y = \phi(\mathbf{x})^T\mathbf{x} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \beta^{-1})$ for all our datapoints. We saw that this formulation could equivalently be expressed as y being random according to a (predictive) distribution

$$p(y \mid \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(\phi(\mathbf{x})^T\mathbf{x}, \beta^{-1}).$$

In such a model, the mean parameter of the Normal distribution is thus modeled by a linear model $\boldsymbol{\mu}(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})$.

In general, we can expect the observations to follow different type of distributions, depending on the type of noise or the type of values target value can take on. E.g., many stochastic processes can not have negative outcomes (e.g. rainfall, waiting times, loan defaults), in which case regressing with a normal distribution would be undesirable!

Gamma distribution $p(y \mid \alpha, \beta)$ for several values of β, with fixed α



For example, when modeling waiting times y in stores, given some input features $\mathbf{x}$, we know we are predicting a quantity that is always positive as one cannot have negative waiting times! An appropriate distribution for such random variables is the gamma distribution. We say that a random variable $y > 0$ is gamma-distributed with shape α and rate β, denoted as $y \sim \text{Gamma}(\alpha, \beta)$, if

$$p(y \mid \alpha, \beta) = \frac{y^{\alpha-1}e^{-\beta y}\beta^{\alpha}}{\Gamma(\alpha)},$$

where Γ denotes the gamma function (whose explicit form we do not need). Let us have a look at optimizing the parameters of the gamma distribution.

2.5p **10a** Suppose we observe a dataset $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ and want to directly model a distribution for y_n, regardless of $\mathbf{x}_n$. I.e. we aim to find a single gamma distribution that models all y_n. Assume α to be given. Give the Maximum Likelihood (ML) estimate of β in terms of α and the data.
Note that, given our modeling assumptions, the solution will not depend on $\mathbf{x}_n$.

3p **10b** Suppose we have the following prior distribution of the β parameter: $\beta \sim \mathsf{Gamma}(a, b)$. Assume hyperparameters α, a, b all to be known. Give the Maximum A-Posteriori (MAP) estimate for β in terms of the data and the known parameters.

2p **10c** **[BONUS]** Suppose now we want our β parameter to depend on some input feature vector $\boldsymbol{\phi}_n := \boldsymbol{\phi}(\mathbf{x}_n)$. Specifically, we want to model the following predictive distribution:

$$p\left(y_n \mid \mathbf{x}_n, \mathbf{w}, \alpha\right) = \mathsf{Gamma}(y_n \mid \alpha, \cosh(\boldsymbol{\phi}_n^T \mathbf{w})).$$

For example, a chain of stores wants to model the customer waiting times as a function of location, time of day and other parameters. Then, y_n correspond to waiting time of the n-th observed customer, while $\boldsymbol{\phi}_n$ would be a vector with information about store location, time of day, etc.

As machine learners, we will make use of our good friend *gradient descent/ascent*. Provide the gradient-based update for model parameters $\mathbf{w}$ with the aim of maximizing the log-likelihood. *Hint: make use of the property that* $\frac{\mathrm{d}}{\mathrm{d}x}\cosh(x) = \sinh(x)$.

Pet Detective

Consider yourself to be a pet detective highly specialized in determining the species of odd looking pets of the "cat" and "dog" variety. In your work, clients come to you with pets of which they are uncertain about their species. Your approach is to collect appearance characteristics (furriness, color, weight, etc.) which you collect in a numeric vector $\mathbf{x} \in \mathbb{R}^d$. Your approach is based on probability theory, and you consider both $\mathbf{x}$ and $c \in \{\text{cat}, \text{dog}\}$ random variables according to some joint distribution $\mathbf{x}, c \sim p(\mathbf{x}, c)$.

1p **11a** You figured that you can best determine pet species c based on the probability for that class given the appearance vector $\mathbf{x}$. You know that any posterior for binary random variables can be written in the form $p(c = \text{dog} \mid \mathbf{x}) = f(a(x))$, with $f(a) = \frac{1}{1+\exp(-a)}$. What is the name for this function f?

1p **11b** Give (or derive) the expression for $a(x)$ in terms of the joint distribution $p(x, c)$.

1p **11c** What are the function values $a(\mathbf{x})$ called, and why?

1p **11d** When modeling joint $p(\mathbf{x}, c)$ with a Gaussian Mixture Model under some assumptions, the function $a(\mathbf{x})$ takes on the shape of a linear function $a(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$. The posterior then thus becomes a generalized linear model! Under what assumptions does this happen?

2p **11e** Suppose you are not interested in the joint $p(\mathbf{x}, c)$, but want to directly model the posterior with a generalized linear model using a database of solved cases $\mathcal{D} = \{(\mathbf{x}_n, c_n)\}_{n=1}^N$. You intend to find the best model parameters $\mathbf{w}$ and b by defining an appropriate loss function and optimize via gradient descent. At the same time you would like to learn which of the measurements in $\mathbf{x}$ are most important, and adapt the loss such that this becomes possible. We want to do feature selection as reducing the number of features could save you time in future investigations!
Which loss should you minimize? *(Answer in words or equations are both fine)*
Are there any hyperparameters to tune?

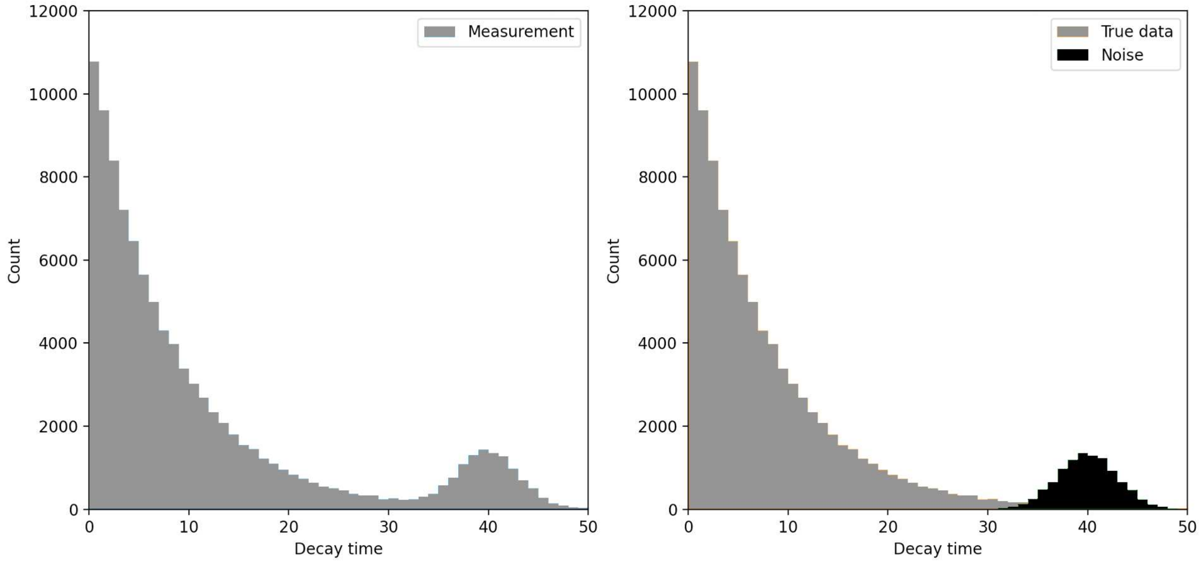
Modeling Muons

Muons are elementary particles, similar to electrons. On earth, muons constantly enter the atmosphere from space and decay into electrons and neutrinos. This time it takes for a muon to decay will be denoted with x, and it follows an exponential distribution

$$\text{Exp}(x|\tau) = \tau e^{-\tau x}.$$

I.e., the decay time x of a muon is random and follows the above distribution specified by half-life time τ. You are a physics student who wants to measure the half-life τ of a muon. In order to do this, you've bought a secondhand detector that measures decay time x of incoming muons and you've let it running all night to get many measurements.

But oh no! The detector is broken. Sometimes it works fine, but other times, it returns random noise. The figure below shows the situation. On the left is your measurement. On the right is what you think might have happened.



Since you studied ML1, you decide to model this as a mixture distribution. You assume that the faulty measurements are normally distributed according to

$$\mathcal{N}(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}.$$

Let $\mathcal{D} = \{x_i\}_{i=1}^N$ denote the collected dataset.

1.5p **12a** Write down the log-likelihood of the data. You can give your answer in terms of $\mathcal{N}(x|\mu, \sigma)$ and Exp$(x|\tau)$ to save yourself some writing. Also, use π_1 to denote the probability for receiving a correct measurement, and π_2 for the probability of a faulty measurement.

1.5p **12b** Write down an expression for the posterior probability that a data point x_n was generated by a faulty measurement as well as the posterior probability that it was created by a good measurement. Again, you can write it in terms of $\mathcal{N}(x|\mu, \sigma)$ and Exp$(x|\tau)$.

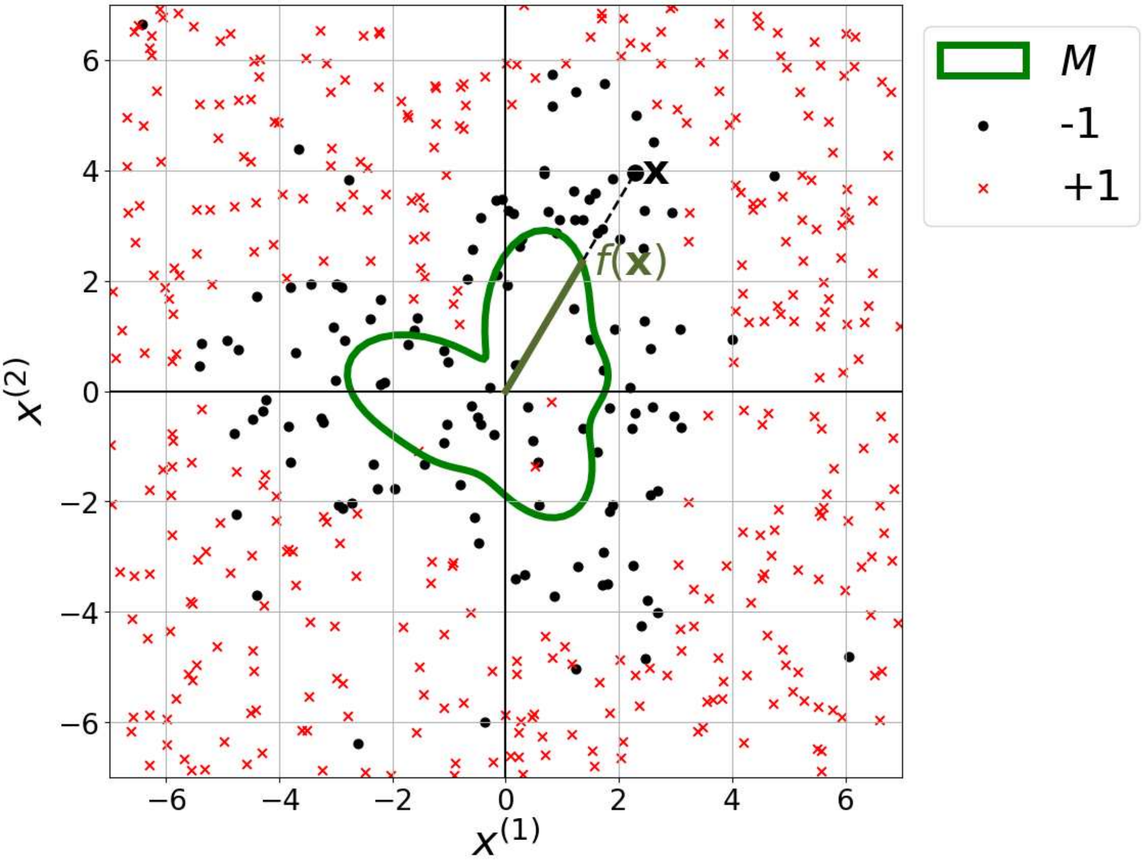
2p **12c** Based on the obtained probabilistic model, you decide to throw away a data point if it is more likely to be noise than to be true data and compute the conditions for when you throw away a measurement x_n. Write the expression in the form of a quadratic inequality i.e. $ax_n^2 + bx_n \geq c$.
Hint: note that a decision boundary based on positive quantities does not change when applying a $\log$ *on both sides. I.e.,* $p_1 \geq p_2 \Leftrightarrow \log p_1 \geq \log p_2$.

2p **12d** The mixture model can be optimized via the Expectation Maximization (EM) algorithm. Derive the M-step equation for the muon half-life τ. Write it in terms of the responsibilities for the faulty class (found in sub question b), which you may denote with the symbol γ.

1p **12e** Let’s assume you successfully derived and applied the EM algorithm to obtain optimal values for π_1, π_2, τ, μ, and σ. Give an expression for the percentage of data you expect to throw away in terms of the relevant model parameters.

Polar Coordinate SVM

[Problem setting] We have a dataset $D = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_N, t_N)\}$ where $\mathbf{x}_n \in \mathbb{R}^2$ and $t_n \in \{-1, +1\}$. The data is centered around the origin and we expect the data to be almost perfectly separable by a decision boundary with a shape similar (up to scaling) to a curve M. See figure below in which the blue points correspond to datapoints for which $t_n = -1$ and the red points correspond to datapoints for which $t_n = +1$.



[Derivation of the maximum margin objective] The boundary M is parameterized by a continuous function $f : \mathbb{R}^2 \to \mathbb{R}$, where $f(\mathbf{x}) \in \mathbb{R}$ *indicates the distance from the origin to the boundary* in the direction of the vector $\mathbf{x} \in \mathbb{R}^2$ (see Figure above). The boundary M is then given by the following set of points

$$M = \left\{ f\left(\begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \right) \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \in \mathbb{R}^2 \ : \ \theta \in [0, 2\pi) \right\} .$$

We want to find a scale $\sigma \geq 0$ such that σM separates our dataset. For simplicity, we measure the distance of a point from the decision boundary in the radial direction, i.e. the signed distance $d(\mathbf{x}, \sigma)$ of a point $\mathbf{x}$ from the boundary σM is given by

$$d(\mathbf{x}, \sigma) = ||\mathbf{x}||_2 - \sigma f(x) .$$

We can use $d(\mathbf{x}, \sigma)$ to define a decision boundary and assign label $t_n = +1$ if $d(\mathbf{x}_n, \sigma) \geq 0$ and $t_n = -1$ otherwise. Two observations are important. Firstly, we have for all correct classifications

$$t_n \, d(\mathbf{x}_n, \sigma) \geq 0 \, .$$

Secondly, the decision boundary does not change when scaling points by a factor α, i.e., $d(\alpha \, \mathbf{x}, \sigma) = 0 \Leftrightarrow \alpha \, d(\mathbf{x}, \sigma) = 0$. This leads to arbitrariness when we want to define a margin. We can get rid of this arbitrary scaling by introducing a variable α which we use to scale the signed distances such that the margin (smallest scaled distance) has value 1. We then have $t_n \, \alpha \, d(\mathbf{x}_n, \sigma) \geq 1$ for all n, which fully written out gives

$$\forall n, \quad \alpha t_n(||\mathbf{x}_n||_2 - \sigma f(\mathbf{x})) \geq 1 \, .$$

Finally, it will be convenient in our derivations later on to apply the change of variable $\beta = \alpha\sigma$ such that

$$\forall n, \quad t_n(\alpha||\mathbf{x}_n||_2 - \beta f(\mathbf{x}_n)) \geq 1 \, .$$

[The objective] We want to maximize the original distances of closest points on the margin to the decision boundary, given by $d(\mathbf{x}_n, \sigma)$. Given the constraint $t_n \, \alpha \, d(\mathbf{x}_n, \sigma) \geq 1$, maximizing $d(\mathbf{x}_n, \sigma)$ implies that we want to minimize α if want the keep the margin at 1. Hence, we will consider the following equivalent problem:

$$\min_{\alpha} \quad \frac{1}{2}\alpha^2, \qquad \text{subject to} \begin{cases} t_n(\alpha||\mathbf{x}_n||_2 - \beta f(\mathbf{x}_n)) & \geq 1 \, , \\ \alpha\beta & \geq 0 \, . \end{cases}$$

Note that we replaced the original condition that $\sigma = \beta/\alpha \geq 0$ with $\alpha\beta \geq 0$, which enforces α and β to agree on their signs and, therefore, σ to be non-negative.

1.5p **13a** Suppose you have solved the optimization and found the optimal values of α and β. What is the size of the margin?

1p **13b** Unfortunately, the points are not exactly separable. To allow for some error in the classification, introduce the slack variables $\{\xi_n\}_{n=1}^N$ and a penalty C for the misclassified points. State the final optimization problem (and explicitly enumerate all the constraints):

2p **13c** Write down the primal Lagrangian. Use the following Lagrange multipliers for each constraint listed above: $\{\lambda_n\}_{n=1}^N$ for the constraints on the margins; $\{\mu_n\}_{n=1}^N$ for those on ξ_n; and γ for the one on $\alpha\beta$. *Indicate which variables are the primal variables and which ones are the dual variables.*

2p **13d** Write down all of the Karush-Kuhn-Tucker (KKT) conditions. How many KKT conditions do we have in total? *(Here we do not consider stationarity as part of the KKT conditions, see next question 13e)*

2p **13e** Optimize the primal Lagrangian with respect to the primal variables. I.e, derive the stationarity conditions. *(answer box continues on the next page)*

↳

⤵

2p **13f** The identities derived from the stationarity conditions can be used to derive expressions for γ. In our derivations we would have to consider three cases:

1. The case $\beta = 0$ and any $\gamma \geq 0$
2. The case $\beta > 0$ and $\gamma > 0$
3. The case $\beta > 0$ and $\gamma = 0$

Using the KKT conditions we can interpret these results. Which of these three cases gives us a sensible classifier, and why can we discard the other two as degenerate solutions?

1p **13g** Given the stationarity conditions, it is possible to show that one derives the dual Lagrangian as follows

$$\hat{L}(\{\lambda_n\}) = -\frac{1}{2}\left(\sum_{n=1}^{N} \lambda_n t_n \|x_n\|_2\right)^2 + \sum_{n=1}^{N} \lambda_n \qquad \text{with constraints } \forall n : \begin{cases} 0 \leq \lambda_n \leq C \\ \sum_{n=1}^{N} \lambda_n t_n f(\mathbf{x}_n) = 0 \end{cases}$$

Give the expression for the kernel function $k(\mathbf{x}_n, \mathbf{x}_m)$ that is effectively used in the above problem.

1p **13h** **[BONUS]** We note that our kernel does not depend on the shape our boundary M, i.e., it does not depend on the function f. Why is this to be expected? How does the boundary M still influence the solution?