Homework 1 Machine Learning

Christoffer Brevik

December 2, 2024

Info to TA/Teacher

Working partners

I have done this assignment independently, but did discuss my answers and general solutions of the tasks with *Isak Tønnersen*, to assurse we had similar answers and logic for this assignment. I do not consider this a formal collaboration as we did not share with each other our papers or show our methods in details, but rather discussed our general solutions and what we concluded with.

Use of language models

As the tasks given here were very similar to examples given in previous lectures (mainly Lecture 3 and 4), I have mostly refrained from using any language models to solve the assignments. Still I have asked **ChatGPT** some questions regarding the homework. I will add the prompts and answers at the end of the document if you are interested in reading trough them

1 Kernel Methods

1.1 Classification of functions as Kernels

1.1.1 Prove that $k(x,y) = (1+xy)^n$ is a kernel

To do this, we need to show that $k(x,y) = (1+xy)^n$ can be expressed as an inner product in a feature space $\mathcal{F}$. Specifically, we aim to find a valid feature mapping $\phi(x)$ so that:

$$k(x,y) = (1+xy)^n = \langle \phi(x), \phi(y) \rangle$$

Using the Binomial Theorem, we can expand $(1+xy)^n$ as follows:

$$(1+xy)^n = \sum_{k=0}^{n} \binom{n}{k} 1^{n-k} (xy)^k$$

Since $1^{n-k} = 1$ for all $k \leq n$, we can simplify this to:

$$(1+xy)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^k$$

This suggests that if we can construct a feature mapping $\phi(x)$ so that each term x^k appears in $\phi(x)$ with the appropriate scaling factor, we may be able to express $k(x,y)$ as an inner product.

To achieve this, we set up our feature mapping $\phi(x)$ as follows:

$$\phi(x) = \sum_{k=0}^{n} \left(\sqrt{\binom{n}{k}} x^k \right) = \left(\sqrt{\binom{n}{0}} x^0, \sqrt{\binom{n}{1}} x^1, \sqrt{\binom{n}{2}} x^2, \ldots, \sqrt{\binom{n}{n}} x^n \right)$$

With this choice of $\phi(x)$, we can compute the inner product $\langle \phi(x), \phi(y) \rangle$:

$$\langle \phi(x), \phi(y) \rangle = \sum_{k=0}^{n} \left(\sqrt{\binom{n}{k}} x^k \right) \left(\sqrt{\binom{n}{k}} y^k \right)$$

Simplifying this, we get:

$$\langle \phi(x), \phi(y) \rangle = \sum_{k=0}^{n} \binom{n}{k} x^k y^k = (1+xy)^n$$

This confirms that we have found a valid feature mapping $\phi(x)$ such that $k(x,y) = (1+xy)^n$ can be expressed as an inner product in a higher-dimensional feature space. Therefore, $k(x,y) = (1+xy)^n$ is indeed a kernel on $\mathcal{X} = \mathbb{R}$.

1.1.2 Disprove that $k(x, y) = xy - 1$ is a kernel

Our task is to prove that $k(x, y) = xy - 1$ is **not** a kernel on $\mathcal{X} = \mathbb{R}$. To do this, we need to show that there are values for $\mathcal{X}$ where $k(x, y) = xy - 1$ cannot be expressed as an inner product in a feature space $\mathcal{F}$.

Here we can use Mercer's Theorem (taken from Lecture 3):

```
Every semi-positive definite symmetric function is a kernel.

Semi-positive definite symmetric functions correspond to a semi-positive
definite symmetric Gram matrix
```

A matrix is positive semi-definite if it's eigenvalues are non-negative. Therefore, to disprove that our function is a kernel, we must find a Gram matrix for our function that has negative eigenvalues. Our function is:

$$k(x, y) = xy - 1$$

NOTE: This solution was found trough experimentation: I first started with a 2x2 matrix with the following values for x and y:

1. $k(1, 1) = 0$
2. $k(1, 0) = -1$
3. $k(-1, -1) = 0$
4. $k(1, 0) = -1$

This can be put this into the following Gram matrix:

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

We can then calculate the eigenvalues of the matrix. by using the formula:

$$|A - \lambda I| = 0$$

$$\begin{vmatrix} -1 - \lambda & 0 \\ 0 & -1 - \lambda \end{vmatrix} = 0$$

$$(-1 - \lambda)^2 = 0$$

$$\lambda) = -1 \vee -1 - \lambda = 0$$

$$\lambda) = -1 \vee \lambda = -1$$

Here we see that the eigenvalues for our Gram matrix aren't positive. Therefore we do not have a semi-positive definite symmetric Gram matrix, teherefore a definite semi-positive function. This goes against Mercers Theorem and proves that the function is **NOT** a kernel.

1.1.3 Prove that $k(x, y) = \min(x, y)$ is a kernel on $X = [0, 1]$

We are tasked with proving that $k(x, y) = \min(x, y)$ is a valid kernel on the set $X = [0, 1]$. To do this, we need to find a feature mapping $\phi(x)$ such that:

$$k(x, y) = \min(x, y) = \langle \phi(x), \phi(y) \rangle$$

Consider the feature mapping $\phi(x)$:

$$\phi(x) = \left(\sqrt{x}, \sqrt{1-x}\right)$$

This is a simple transformation that will allow us to capture the comparison between x and y in a higher-dimensional feature space.

Now, we compute the inner product $\langle \phi(x), \phi(y) \rangle$:

$$\langle \phi(x), \phi(y) \rangle = \left(\sqrt{x}, \sqrt{1-x}\right) \cdot \left(\sqrt{y}, \sqrt{1-y}\right)$$

This inner product simplifies to:

$$\langle \phi(x), \phi(y) \rangle = \sqrt{x}\sqrt{y} + \sqrt{1-x}\sqrt{1-y}$$

We now compare the result with the original function $k(x, y) = \min(x, y)$. It is known that for $x, y \in [0, 1]$, the function $\min(x, y)$ behaves as:

$$\min(x, y) = \sqrt{x}\sqrt{y} \quad \text{if} \quad x = y$$

With $\sqrt{1-x}\sqrt{1-y}$, the comparison reflects the minimum of the two values. As this is a valid kernel and feature space, we conclude that $k(x, y) = \min(x, y)$ can indeed be expressed as an inner product in the feature space $\mathcal{F}$. We have therefore proved that:

$$k(x, y) = \min(x, y)$$

is a valid kernel on $X = [0, 1]$.

1.2 Kernel SVM for Classification

1.2.1 0-1 loss with Hinge-loss as aproximation

The 0-1 loss function, $\ell_{0-1}(yf(x)) = \mathbb{I}[yf(x) < 0]$, directly penalizes misclassifications by assigning a loss of 1 whenever $yf(x) < 0$ and 0 otherwise. Tue 0-1 loss has suffers from non-convexity and is discontinuous, making it difficult to optimize using gradient-based methods, as it can lead to local minima.

On the other hand, the hinge loss $\ell_{\text{hinge}}(yf(x)) = \max(0, 1 - yf(x))$ is a convex function that provides an upper bound to the 0-1 loss. Furthermore it is also continuous. Therefore approximating the 0-1 loss using hinge loss can prove very beneficial.

Still, the hinge loss is not smooth, meaning it is not differentiable at all points. An alternative loss function which is both convex, continuous and smooth that upper-bounds the 0-1 loss is the **logistic loss function**. This function can therefore serve as a good alternative to the hinge-loss.

1.2.2 Write down Lagrangian function of function

As this task was worded as "Write down" and that an example of a Lagrangian function based on a very similar function in Lecture 3, I will keep my explanation brief. For this constrained optimization problem, we construct a Lagrangian function by introducing Lagrange multipliers for each constraint. Let:

- $\alpha_i \geq 0$ be the Lagrange multiplier for the constraint $y_i(\mathbf{w}^T\phi(x_i)) \geq 1 - \xi_i$.
- $\mu_i \geq 0$ be the Lagrange multiplier for the constraint $\xi_i \geq 0$.

The Lagrangian function L is then given by:

$$L(\mathbf{w}, \boldsymbol{\xi}, \alpha, \mu) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\left(y_i(\mathbf{w}^T\phi(x_i)) - 1 + \xi_i\right) - \sum_{i=1}^{N}\mu_i\xi_i.$$

$$= \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i y_i(\mathbf{w}^T\phi(x_i)) + \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i\xi_i - \sum_{i=1}^{N}\mu_i\xi_i.$$

While this is satisfactory, we see that we can combine the terms containing ξ_i:

$$\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\xi_i - \sum_{i=1}^{N}\mu_i\xi_i = \sum_{i=1}^{N}\xi_i(1 - \alpha_i - \mu_i).$$

We therefore end with the final Lagrangian function of (1.1):

$$L(\mathbf{w}, \boldsymbol{\xi}, \alpha, \mu) = \frac{\lambda}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N}\alpha_i y_i(\mathbf{w}^T\phi(x_i)) + \sum_{i=1}^{N}\alpha_i + \sum_{i=1}^{N}\xi_i(1 - \alpha_i - \mu_i).$$

1.2.3 Problem 1.6

To derive the dual problem of (1.1), we can start with the Lagrangian function we derived in the previous task:

$$L(\mathbf{w}, \boldsymbol{\xi}, \alpha, \mu) = \frac{\lambda}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i y_i (\mathbf{w}^T \phi(x_i)) + \sum_{i=1}^{N} \alpha_i + \sum_{i=1}^{N} \xi_i (1 - \alpha_i - \mu_i).$$

We now find the maximum value for $\mathbf{w}$, $\hat{\mathbf{w}}$, by finding out when the derivative is zero:

$$\frac{\partial L}{\partial \mathbf{w}} L(\mathbf{w}, \boldsymbol{\xi}, \alpha, \mu) = 0$$

$$\frac{\partial L}{\partial \mathbf{w}} (\frac{\lambda}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i y_i (\mathbf{w}^T \phi(x_i)) + \sum_{i=1}^{N} \alpha_i + \sum_{i=1}^{N} \xi_i (1 - \alpha_i - \mu_i)) = 0.$$

$$\lambda \hat{\mathbf{w}} - \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) = 0.$$

Solving for $\mathbf{w}$, we get:

$$\mathbf{w} = \frac{1}{\lambda} \sum_{i=1}^{N} \alpha_i y_i \phi(x_i).$$

Substituting $\hat{\mathbf{w}} = \frac{1}{\lambda} \sum_{i=1}^{N} \alpha_i y_i \phi(x_i)$ back into the Lagrangian, we have:

$$L(\hat{\mathbf{w}}, \boldsymbol{\xi}, \alpha, \mu) = \frac{\lambda}{2}\|\hat{\mathbf{w}}\|^2 - \sum_{i=1}^{N} \alpha_i y_i (\hat{\mathbf{w}}^T \phi(x_i)) + \sum_{i=1}^{N} \alpha_i + \sum_{i=1}^{N} \xi_i (1 - \alpha_i - \mu_i)$$

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2\lambda} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j).$$

Thus, the dual problem becomes:

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2\lambda} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j),$$

subject to $0 \leq \alpha_i \leq 1$ for all i.

1.2.4 Prediction function

To express the prediction function $f(x) = \text{sign}(\hat{w}^\top \phi(x))$ using the kernel and the solutions of the dual problem, we proceed as follows:

From the last question we had:

$$\hat{w} = \frac{1}{\lambda} \sum_{i=1}^{N} \alpha_i y_i \phi(x_i)$$

where α_i are the dual variables from the solution of the dual problem. We can now substitute $\hat{w}$ into the prediction function, getting:

$$f(x) = \text{sign}(\hat{w}^\top \phi(x)) = \text{sign}\left(\left(\frac{1}{\lambda} \sum_{i=1}^{N} \alpha_i y_i \phi(x_i)\right)^\top \phi(x)\right)$$

Since $\phi(x_i)^\top \phi(x) = k(x_i, x)$, we can rewrite the expression so that it doesn't contain the feature map ϕ:

$$f(x) = \text{sign}\left(\frac{1}{\lambda} \sum_{i=1}^{N} \alpha_i y_i k(x_i, x)\right)$$

2 Exponential Families

We consider the exponential family, given by the probability density function 2.1:

$$p(x|\eta) = h(x)\exp(\eta^T T(x) - A(\eta)),$$

where $T(x)$ is a sufficient statistic, and $A(\eta) = \log \int h(x)e^{\eta^T T(x)}\,dx$ is the partition function. We are asked to verify 2.2:

$$\frac{\partial}{\partial \eta_i} A(\eta) = \mathbb{E}_{p(x|\eta)}[T_i(x)],$$

where $T_i(x)$ is the i-th component of $T(x)$. We are also tasked with verifying 2.3:

$$\frac{\partial^2}{\partial \eta_i \partial \eta_j} A(\eta) = \mathrm{Cov}_{p(x|\eta)}[T_i(x), T_j(x)].$$

2.1 Verify (2.2)

To prove 2.2, we can just differentiate function A, with respect to η:

$$A(\eta) = \log \int h(x)\exp(\eta^T T(x))\,dx$$

Using the chain rule, and stating that $A(\eta) = \log(g(\eta))$ where $g(\eta) = \int h(x)\exp(\eta^T T(x))\,dx$. We get the following:

$$\frac{\partial A(\eta)}{\partial \eta} = \frac{1}{g(\eta)}\frac{\partial g(\eta)}{\partial \eta}.$$

We then Compute $\frac{\partial g(\eta)}{\partial \eta}$:

$$\frac{\partial g(\eta)}{\partial \eta} = \frac{\partial}{\partial \eta}\int h(x)\exp(\eta^T T(x))\,dx.$$

As we only have the exponent containing η, and the integration is with respects to x, we can move the derivative into the integration, getting:

$$\frac{\partial g(\eta)}{\partial \eta} = \int h(x)\frac{\partial}{\partial \eta}\exp(\eta^T T(x))\,dx.$$

The derivative of $\exp(\eta^T T(x))$ with respect to η is $T(x)\exp(\eta^T T(x))$, we therefore get:

$$\frac{\partial g(\eta)}{\partial \eta} = \int h(x)T(x)\exp(\eta^T T(x))\,dx.$$

From here we can actually take a shortcut. If we go back to our original formula, we can replace $A(\eta)$ with $log(g(\eta))$:

$$p(x|\eta) = h(x)\exp(\eta^T T(x) - A(\eta))$$

$$p(x|\eta) = h(x)\exp(\eta^T T(x) - log(g(\eta)))$$

$$p(x|\eta) = h(x)\frac{\exp(\eta^T T(x))}{\exp(log(g(\eta))}$$

$$p(x|\eta) = h(x)\frac{\exp(\eta^T T(x))}{g(\eta)},$$

$$g(\eta) = h(x)\frac{\exp(\eta^T T(x))}{p(x|\eta)},$$

We can now go back to our derivative and substitute our derivatives in:

$$\frac{\partial A(\eta)}{\partial \eta} = \frac{1}{g(\eta)}\frac{\partial g(\eta)}{\partial \eta}.$$

$$\frac{\partial A(\eta)}{\partial \eta} = \frac{1}{\int h(x)\exp(\eta^T T(x'))\,dx'} \int h(x)T(x)\exp(\eta^T T(x))\,dx.$$

$$\frac{\partial A(\eta)}{\partial \eta} = \int T(x) \frac{h(x)\exp(\eta^T T(x))}{\int h(x)\exp(\eta^T T(x'))\,dx'}\,dx.$$

Note that from $p(x|\eta) = h(x)\frac{\exp(\eta^T T(x))}{g(\eta)}$ and $g(\eta) = \int h(x)\exp(\eta^T T(x))\,dx$ we know:

$$p(x|\eta) = h(x)\frac{\exp(\eta^T T(x))}{\int h(x)\exp(\eta^T T(x))\,dx}$$

We see that our derivative of A contains this function, we can therefore simplify our function to:

$$\frac{\partial A(\eta)}{\partial \eta} = \int T(x)p(x|\eta)dx.$$

Now, since $p(x|\eta)$ represents the probability density of x given η , this integral is the expected value of $T(x)$ with respect to the distribution $p(x|\eta)$. Therefore, we can conclude:

$$\frac{\partial A(\eta)}{\partial \eta} = \int T(x)p(x|\eta)dx = \mathbb{E}_{p(x;\eta)}[T(x)].$$

which verifies equation (2.2).

2.2 Verify (2.3)

Now, let's find $\frac{\partial^2}{\partial\eta_i\partial\eta_j}A(\eta)$ by differentiating $\frac{\partial}{\partial\eta_i}A(\eta)$ with respect to η_j. For this we can use the result from 2.1:

$$\frac{\partial^2}{\partial\eta_i\partial\eta_j}A(\eta) = \frac{\partial}{\partial\eta_j}(\frac{\partial}{\partial\eta_i}A(\eta)) = \frac{\partial}{\partial\eta_j}\mathbb{E}_{p(x|\eta)}[T_i(x)].$$

Since we know that $\mathbb{E}_{p(x|\eta)}[T_i(x)] = \int T_i(x)p(x|\eta)\,dx$, we can use this:

$$\frac{\partial^2}{\partial\eta_i\partial\eta_j}A(\eta) = \frac{\partial}{\partial\eta_j}\int T_i(x)p(x|\eta)\,dx$$

We see that the integral is with respect to x, that the derivative is with respect to η and that $p(x|\eta)$ is the only function containing η. We can therefore move the derivative inside the integral:

$$\frac{\partial^2}{\partial\eta_i\partial\eta_j}A(\eta) = \int T_i(x)\frac{\partial}{\partial\eta_j}p(x|\eta)\,dx$$

We will now calculate $\frac{\partial}{\partial\eta_j}p(x|\eta)$:

$$\frac{\partial}{\partial\eta_j}p(x|\eta) = \frac{\partial}{\partial\eta_j}\big(h(x)\exp(\eta^T T(x) - A(\eta))\big)$$

$$\frac{\partial}{\partial\eta_j}p(x|\eta) = h(x)\frac{\partial}{\partial\eta_j}\exp(\eta^T T(x) - A(\eta))$$

Here we can use the chain rule:

$$\frac{\partial}{\partial\eta_j}p(x|\eta) = h(x) - \exp(\eta^T T(x) - A(\eta))\frac{\partial}{\partial\eta_j}(\eta^T T(x) - A(\eta))$$

We see that this function actually contains our original $p(x|\eta)$:

$$\frac{\partial}{\partial\eta_j}p(x|\eta) = h(x) - \exp(\eta^T T(x) - A(\eta))(T_j(x) - \mathbb{E}[T_j(X)])$$

$$\frac{\partial}{\partial\eta_j}p(x|\eta) = p(x|\eta)(T_j(x) - \mathbb{E}[T_j(X)])$$

With $\frac{\partial}{\partial\eta_j}p(x|\eta)$ calculated, we can now insert this into $\frac{\partial^2}{\partial\eta_i\partial\eta_j}A(\eta)$:

$$\frac{\partial^2}{\partial\eta_i\partial\eta_j}A(\eta) = \int T_i(x)\frac{\partial}{\partial\eta_j}p(x|\eta)\,dx$$

$$\frac{\partial^2}{\partial\eta_i\partial\eta_j}A(\eta) = \int T_i(x)(p(x|\eta)(T_j(x) - \mathbb{E}[T_j(X)]))\,dx$$

Now, we expand the integral:

$$\frac{\partial^2}{\partial\eta_i\partial\eta_j}A(\eta) = \int T_i(x)p(x|\eta)T_j(x)\,dx - \mathbb{E}_{p(x|\eta)}[T_j(x)]\int T_i(x)p(x|\eta)\,dx.$$

Here we can make the following insights:

1. The first term is simply the expected value of the product of $T_i(x)$ and $T_j(x)$:

$$\mathbb{E}_{p(x|\eta)}[T_i(x)T_j(x)].$$

2. The second term involves the product of the expected values of $T_i(x)$ and $T_j(x)$:

$$\mathbb{E}_{p(x|\eta)}[T_i(x)]\mathbb{E}_{p(x|\eta)}[T_j(x)].$$

Thus, the second derivative becomes:

$$\frac{\partial^2}{\partial \eta_i \partial \eta_j} A(\eta) = \mathbb{E}_{p(x|\eta)}[T_i(x)T_j(x)] - \mathbb{E}_{p(x|\eta)}[T_i(x)]\mathbb{E}_{p(x|\eta)}[T_j(x)].$$

We know that the definition of covariance is:

$$\mathrm{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$$

This is precisely what we have. We can therefore conclude with:

$$\frac{\partial^2}{\partial \eta_i \partial \eta_j} A(\eta) = \mathrm{Cov}_{p(x|\eta)}[T_i(x), T_j(x)],$$

which verifies equation (2.3).

3 Maximum Likelyhood Estimators

3.1 Finding the MLE estimators $\hat{\mu}_{\mathbf{ML}}$ and $\hat{\Sigma}_{\mathbf{ML}}$

We see that our multivariate Gaussian distribution:

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right).$$

Belongs to the exponential family:

$$p(x|\eta) = h(x)\exp(\eta^T T(x) - A(\eta)),$$

We can therefore use the the log-likelihood to remove the exponent, as in Lecture 4, and use the derivatives to find the local maxima. For our distribution the log-likelihood function for the dataset is:

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{n} \log p(\mathbf{x}_i|\boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

Substituting the probability density function, we get:

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{n}\left(-\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})\right).$$

This simplifies to:

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu}).$$

Estimating $\hat{\mu}$

To estimate the MLE of μ, we start by the taking derivative with respects of μ, and finding out when this is 0:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = 0$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = \frac{\partial}{\partial \boldsymbol{\mu}}(-\frac{nd}{2}\log(2\pi)) - \frac{\partial}{\partial \boldsymbol{\mu}}(\frac{n}{2}\log|\boldsymbol{\Sigma}|) - \frac{\partial}{\partial \boldsymbol{\mu}}(\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})).$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = -\frac{\partial}{\partial \boldsymbol{\mu}}(\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})).$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = -\frac{1}{2}\sum_{i=1}^{n}\frac{\partial}{\partial \boldsymbol{\mu}}((\mathbf{x}_i-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})).$$

We can now look at the inner derivative, by separating this into it's own function:

$$f(\boldsymbol{\mu}) = (\mathbf{x}_i-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})$$

$$f(\boldsymbol{\mu}) = \mathbf{x}_i^\top \boldsymbol{\Sigma}^{-1}\mathbf{x}_i - 2\mathbf{x}_i^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$$

$$\frac{\partial f}{\partial \boldsymbol{\mu}} = \frac{\partial}{\partial \boldsymbol{\mu}}\left(\mathbf{x}_i^\top \boldsymbol{\Sigma}^{-1}\mathbf{x}_i\right) - 2\frac{\partial}{\partial \boldsymbol{\mu}}\left(\mathbf{x}_i^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right) + \frac{\partial}{\partial \boldsymbol{\mu}}\left(\boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right)$$

$$\frac{\partial f}{\partial \boldsymbol{\mu}} = -2\boldsymbol{\Sigma}^{-1}\mathbf{x}_i + 2\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$$

$$\frac{\partial f}{\partial \boldsymbol{\mu}} = 2\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}-\mathbf{x}_i)$$

We can now add this back in our original formula:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = -\frac{1}{2}\sum_{i=1}^{n}\frac{\partial f}{\partial \boldsymbol{\mu}}.$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = -\frac{1}{2}\sum_{i=1}^{n} 2\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{x}_i).$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = -\sum_{i=1}^{n} \frac{(\boldsymbol{\mu} - \mathbf{x}_i)}{\Sigma}$$

Setting this to zero gives:

$$0 = -\sum_{i=1}^{n} \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - \mathbf{x}_i)$$

$$\sum_{i=1}^{n}(\boldsymbol{\mu} - \mathbf{x}_i) = 0$$

$$n\boldsymbol{\mu} - \sum_{i=1}^{n} \mathbf{x}_i = 0$$

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$

Thus, the Maximum Likelihood Estimator for $\boldsymbol{\mu}$ is the sample mean:

$$\hat{\mu}_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$$

3.1.1 Estimating $\hat{\Sigma}$

We will now do the same as for $\hat{\mu}$, but derivate with respects to Σ instead:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}} = 0$$

Differentiating the log-likelihood function with respect to $\boldsymbol{\Sigma}$, we get:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}} = \frac{\partial}{\partial \boldsymbol{\Sigma}}(-\frac{nd}{2}\log(2\pi)) - \frac{\partial}{\partial \boldsymbol{\Sigma}}(\frac{n}{2}\log|\boldsymbol{\Sigma}|) - \frac{\partial}{\partial \boldsymbol{\Sigma}}(\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})).$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}} = \frac{\partial}{\partial \boldsymbol{\Sigma}}\left(-\frac{n}{2}\log|\boldsymbol{\Sigma}|\right) - \frac{\partial}{\partial \boldsymbol{\Sigma}}\left(\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right)$$

The first term simplifies to:

$$\frac{\partial}{\partial \boldsymbol{\Sigma}}\left(-\frac{n}{2}\log|\boldsymbol{\Sigma}|\right) = -\frac{n}{2}\boldsymbol{\Sigma}^{-1}$$

For the second term, we differentiate:

$$\frac{\partial}{\partial \boldsymbol{\Sigma}}\left(\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right)$$

$$\frac{\partial}{\partial \boldsymbol{\Sigma}}\left((\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right) = -\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}$$

We now add this back to the original formula:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}} = \frac{\partial}{\partial \boldsymbol{\Sigma}}\left(-\frac{n}{2}\log|\boldsymbol{\Sigma}|\right) - \frac{\partial}{\partial \boldsymbol{\Sigma}}\left(\frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}} = -\frac{n}{2}\boldsymbol{\Sigma}^{-1} + \frac{1}{2}\sum_{i=1}^{n}\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}.$$

Setting this to zero:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\Sigma}} = 0$$

$$-\frac{n}{2}\boldsymbol{\Sigma}^{-1} + \frac{1}{2}\sum_{i=1}^{n} \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} = 0$$

$$\frac{1}{\Sigma}\left(-\frac{n}{2} + \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top\right) = 0$$

$$-\frac{n}{2} + \frac{1}{2}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top = 0$$

$$\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top = n$$

Multiplying by $\boldsymbol{\Sigma}$, we arrive at:

$$n\boldsymbol{I} = \sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top$$

The Maximum Likelihood Estimator for $\boldsymbol{\Sigma}$ is:

$$\hat{\Sigma}_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top.$$

3.2 Computing $\mathbb{E}[\hat{\mu}_{\mathbf{ML}}]$ and $\mathbb{E}[\hat{\Sigma}_{\mathbf{ML}}]$, and checking if they are unbiased

To check if $\mathbb{E}[\hat{\mu}_{\text{ML}}]$ and $\mathbb{E}[\hat{\Sigma}_{\text{ML}}]$ are unbiased, we need to compare the expectations and $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and see if these are equal (given by the hint in the assignment).

3.2.1 Expectation of $\hat{\mu}_{\mathbf{ML}}$

Recall that the maximum likelihood estimator for $\boldsymbol{\mu}$ is given by:

$$\hat{\mu}_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$$

where $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Since $\mathbb{E}[\mathbf{x}_i] = \boldsymbol{\mu}$, we can compute $\mathbb{E}[\hat{\mu}_{\text{ML}}]$ as follows:

$$\mathbb{E}[\hat{\mu}_{\text{ML}}] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[\mathbf{x}_i] = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\mu} = \boldsymbol{\mu}$$

Thus, $\mathbb{E}[\hat{\mu}_{\text{ML}}] = \boldsymbol{\mu}$, which shows that $\hat{\mu}_{\text{ML}}$ is an unbiased estimator of $\boldsymbol{\mu}$.

3.2.2 Expectation of $\hat{\Sigma}_{\mathbf{ML}}$

The maximum likelihood estimator for $\boldsymbol{\Sigma}$ is given by:

$$\hat{\Sigma}_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mu}_{\text{ML}})(\mathbf{x}_i - \hat{\mu}_{\text{ML}})^\top$$

To determine if this estimator is unbiased, we can express $\mathbf{x}_i = \boldsymbol{\mu} + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \boldsymbol{\Sigma})$. Substituting this into $\hat{\Sigma}_{\text{ML}}$, we get:

$$\hat{\Sigma}_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}\left((\mathbf{x}_i - \boldsymbol{\mu}) - (\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})\right)\left((\mathbf{x}_i - \boldsymbol{\mu}) - (\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})\right)^\top$$

$$\hat{\Sigma}_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}\left((\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top - (\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})(\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})^\top\right)$$

$$\hat{\Sigma}_{\text{ML}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top - \frac{1}{n}\sum_{i=1}^{n}(\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})(\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})^\top$$

We now look at the expectation, $\mathbb{E}[\hat{\Sigma}_{\text{ML}}]$, so we get:

$$\mathbb{E}[\hat{\Sigma}_{\text{ML}}] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top - \frac{1}{n}\sum_{i=1}^{n}(\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})(\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})^\top\right]$$

$$\mathbb{E}[\hat{\Sigma}_{\text{ML}}] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top\right] - \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})(\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})^\top\right]$$

Taking expectations of each term, we get:

- For the first term, $\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top\right] = \boldsymbol{\Sigma}$.
- For the second term, since $\hat{\mu}_{\text{ML}}$ is an average of n i.i.d. samples from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, we have $\mathbb{E}[(\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})(\hat{\mu}_{\text{ML}} - \boldsymbol{\mu})^\top] = \frac{1}{n}\boldsymbol{\Sigma}$.

Thus,

$$\mathbb{E}[\hat{\Sigma}_{\text{ML}}] = \boldsymbol{\Sigma} - \frac{1}{n}\boldsymbol{\Sigma} = \frac{n-1}{n}\boldsymbol{\Sigma}$$

This shows that $\hat{\Sigma}_{\text{ML}}$ is a biased estimator, as $\mathbb{E}[\hat{\Sigma}_{\text{ML}}] \neq \boldsymbol{\Sigma}$.

3.3 Showing that $\mathbb{E}\left[\|\hat{\mu}_{\mathbf{ML}} - \mu\|^2\right] = \frac{\operatorname{Tr}\Sigma}{n}$

We recognize our left side as the expression for the expected squared error:

$$\mathbb{E}\left[\|\hat{\mu}_{\mathrm{ML}} - \mu\|^2\right].$$

We can rewrite this as:

$$\mathbb{E}\left[\|\hat{\mu}_{\mathrm{ML}} - \mu\|^2\right] = \mathbb{E}\left[(\hat{\mu}_{\mathrm{ML}} - \mu)^\top(\hat{\mu}_{\mathrm{ML}} - \mu)\right].$$

We can now add our definition of $\hat{\mu}_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ from 3.1.1:

$$\mathbb{E}\left[\|\hat{\mu}_{\mathrm{ML}} - \mu\|^2\right] = \mathbb{E}\left[(\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i - \mu)^\top(\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i - \mu)\right].$$

Expanding the inner product, we get:

$$\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i - \mu\right)^\top \left(\frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i - \mu\right)\right]$$

$$\mathbb{E}\left[\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\mathbf{x}_i - \mu)^\top(\mathbf{x}_j - \mu)\right].$$

We can now look at the different values of i and j:

- When $i = j$
$$\mathbb{E}\left[(\mathbf{x}_i - \mu)^\top(\mathbf{x}_i - \mu)\right] = \operatorname{Tr}(\Sigma).$$
- When $i \neq j$, $\mathbf{x}_i$ and $\mathbf{x}_j$ are independent, we have:
$$\mathbb{E}\left[(\mathbf{x}_i - \mu)^\top(\mathbf{x}_j - \mu)\right] = 0.$$

Substituting these results, we are left with:

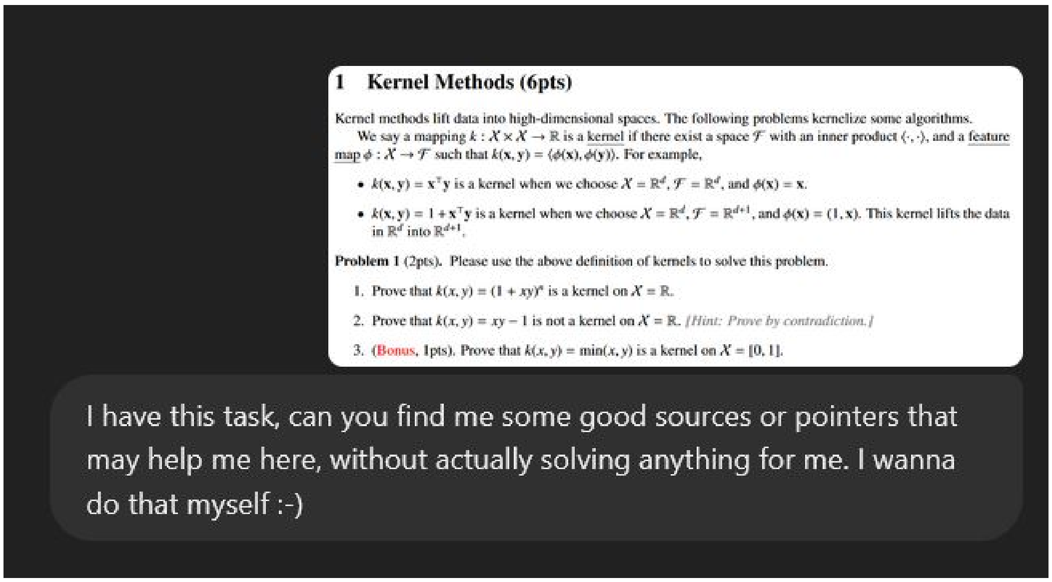
$$\mathbb{E}\left[\|\hat{\mu}_{\mathrm{ML}} - \mu\|^2\right] = \frac{1}{n^2} \cdot \sum_{i=1}^{n} \cdot \operatorname{Tr}(\Sigma) = \frac{\operatorname{Tr}(\Sigma)}{n}.$$

Photos from ChatGPT

Apart from googling independently and looking trough the lectures I did ask **ChatGPT** for help with finding similar tasks and sources. Most of these sources were either ones I've already found or not that useful, but some did prove central in my solution. Below are the 5 promts I did send during this project:

Task 1 Part 1

My prompt



1 Kernel Methods (6pts)

Kernel methods lift data into high-dimensional spaces. The following problems kernelize some algorithms.

We say a mapping $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel if there exist a space $\mathcal{F}$ with an inner product $\langle \cdot, \cdot \rangle$, and a feature map $\phi : \mathcal{X} \to \mathcal{F}$ such that $k(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$. For example,

- $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$ is a kernel when we choose $\mathcal{X} = \mathbb{R}^d$, $\mathcal{F} = \mathbb{R}^d$, and $\phi(\mathbf{x}) = \mathbf{x}$.
- $k(\mathbf{x}, \mathbf{y}) = 1 + \mathbf{x}^\top \mathbf{y}$ is a kernel when we choose $\mathcal{X} = \mathbb{R}^d$, $\mathcal{F} = \mathbb{R}^{d+1}$, and $\phi(\mathbf{x}) = (1, \mathbf{x})$. This kernel lifts the data in $\mathbb{R}^d$ into $\mathbb{R}^{d+1}$.

Problem 1 (2pts). Please use the above definition of kernels to solve this problem.

1. Prove that $k(x, y) = (1 + xy)^n$ is a kernel on $\mathcal{X} = \mathbb{R}$.
2. Prove that $k(x, y) = xy - 1$ is not a kernel on $\mathcal{X} = \mathbb{R}$. *[Hint: Prove by contradiction.]*
3. (Bonus, 1pts). Prove that $k(x, y) = \min(x, y)$ is a kernel on $\mathcal{X} = [0, 1]$.

Response

Here are some useful concepts and resources to help you approach this task:

1. **Mercer's Theorem:** This is a key result in kernel theory. It states that a function $k(x, y)$ can be considered a valid kernel if it is symmetric and positive semi-definite. Reviewing Mercer's theorem might help you understand the requirements for proving whether a function is a kernel.
2. **Feature Mapping:** Since kernels work by implicitly mapping inputs into a higher-dimensional space, understanding how to construct or identify feature maps (i.e., the function $\phi(x)$) is essential. The goal is to rewrite $k(x, y)$ as $\langle \phi(x), \phi(y) \rangle$ for some ϕ. A classic reference is "*Pattern Recognition and Machine Learning*" by Christopher Bishop, which covers kernels and feature maps.
3. **Kernel Types and Properties:**
   - **Polynomial Kernels**: Your first question involves a polynomial form, $(1 + xy)^n$. Polynomial kernels are well-known, and seeing how they satisfy kernel properties could be useful.
   - **Non-Kernel Examples:** For the second part, proving something isn't a kernel can be approached through contradiction by showing it doesn't satisfy positive semi-definiteness. For guidance on non-kernel examples, look up resources discussing non-kernels in machine learning.
4. **Bounding and Min Functions:** The third (bonus) part involves min functions. You might find useful insights in sections of books on kernels that discuss the "min kernel" or "bounded similarity functions." These are commonly discussed in books on kernel methods like "*Kernel Methods for Pattern Analysis*" by John Shawe-Taylor and Nello Cristianini.

Let me know if you need more specific references or additional guidance on any of these points. Good luck!

Task 1 Part 2

I have this task:

"

Problem 1(2pts).Please use the above definition of kernels to solve this problem.1. Prove thatk(x,y)=(1+xy)nis a kernel onX=R.2. Prove thatk(x,y)=xy−1 is not a kernel onX=R.[Hint: Prove by contradiction.]3. (Bonus, 1pts). Prove thatk(x,y)=min(x,y) is a kernel onX=[0,1].Problem 2(Kernel SVM for Classification, 4pts).Given a training dataset{(xi,yi)} Ni=1, wherexi∈Rdandyi∈{±1}. Letφ:Rd→Rmbe a feature map. Considerthe following Soft SVM problem:minw∈Rm,ξ∈RNλ2‖w‖2+NXi=1ξis.t. ξi≥0,yiw⊤φ(xi)≥1−ξi.(1.1)1. Hinge lossℓlin(y f(x))=max(0,1−y f(x)) is used in the formulation of (1.1). Please explain why we oftenuse Hinge loss to approximate 0-1 lossℓ0−1(y f(x))=I[y f(x)<0] in practice. Give another feasible functionto approximate 0-1 loss.[Hint: Consider its convexity and whether it upperbounds 0-1 loss.]1

2. Write down the Lagrangian function of (1.1).3. Derive the dual problem of (1.1) using the kernelk(x,y)=φ(x)⊤φ(y) instead of the feature mapφ.Thefeaturemapφisnotallowedtoappearintheresult.4. Letˆw,ˆξbe the solution of (1.1). Express the prediction functionf(x)=sign(ˆw⊤φ(x)) using the kernel andthe solutions of the dual problem.Thefeaturemapφisnotallowedtoappearintheresult

"

Please give me some sources or pointers that may help me solve this task but DON'T DO ANY CALCULATIONS

Response

For solving the task, I would recommend breaking it down into a few key topics. Here are some useful sources and pointers to guide you through each part:

Part 1: Proving Kernel Properties

1. **Kernel Definition and Properties:**
   - Understand the definition of a **positive semi-definite kernel**. A kernel $k(x, y)$ is positive semi-definite if for any finite set of points $\{x_1, x_2, ..., x_n\}$, the corresponding kernel matrix K (where $K_{ij} = k(x_i, x_j)$) is positive semi-definite. This is the main tool to show whether a function is a valid kernel.
   - **Proving $k(x, y) = (1 + xy)^n$ is a kernel**: This can be done by showing that it is positive semi-definite, often using the fact that polynomials in x and y can be written as inner products in some feature space.
     - A good place to start is the proof techniques for **polynomial kernels**. You may find explanations of this in standard **Machine Learning** books like *Pattern Recognition and Machine Learning* by Christopher Bishop or *The Elements of Statistical Learning* by Hastie, Tibshirani, and Friedman.
2. **Proving $k(x, y) = xy - 1$ is not a kernel**:
   - Use the **contradiction method** and try to find a counterexample where the kernel fails to be positive semi-definite. You can find examples of how this is done in texts like *Learning with Kernels* by Bernhard Schölkopf and Alexander J. Smola.
   - For this part, focus on the property of kernels and look for a situation where the matrix formed by this kernel function does not satisfy the positive semi-definite condition.

Part 2: Kernel SVM for Classification

1. Hinge Loss Approximation:
   - The Hinge loss is commonly used in SVMs as it approximates the 0-1 loss, which is not convex. The idea is that Hinge loss penalizes predictions that are on the wrong side of the decision boundary and is convex, which makes it easier to optimize.
   - To understand the intuition behind this, check out *Support Vector Machines for Pattern Classification* by Shigeo Abe. This will give you an overview of why convex losses like Hinge are preferred in practice.
2. Lagrangian of the Soft SVM Problem:
   - For the Lagrangian formulation of the Soft SVM problem, you can look into the SVM section of any standard machine learning textbook, like *Pattern Recognition and Machine Learning* by Christopher Bishop, or *The Elements of Statistical Learning* by Hastie et al.
   - The Lagrangian will involve introducing the Lagrange multipliers for the constraints $\xi_i \geq 0$ and $y_i w^\top \phi(x_i) \geq 1 - \xi_i$.
3. Dual Problem of SVM with Kernels:
   - The dual of the SVM problem can be derived using the kernel trick. Here, you'll express the problem in terms of the kernel function, where you replace the feature map $\phi(x)$ with $k(x, y) = \phi(x)^\top \phi(y)$.
   - Learning with Kernels by Schölkopf and Smola is a great resource here as it provides an in-depth explanation of the dual formulation of SVM with kernels and the derivation of the dual problem.
4. Prediction Function Using the Kernel:
   - After solving the dual problem, the prediction function in terms of the kernel can be derived. The prediction function will involve the dual variables (Lagrange multipliers) and the kernel function.
   - This part can be understood by looking into how the dual solution allows you to express the SVM's decision boundary in terms of the kernel, as described in *The Elements of Statistical Learning* and *Learning with Kernels*.

Task 2 Part 1

My prompt

Please don't use any book references but websites instead :-)

Response

Part 1: Proving Kernel Properties

1. **Understanding Kernels and Positive Semi-Definiteness:**
   - Wikipedia on Positive Semi-Definite Kernels – This page explains the concept of positive semi-definite kernels, which is crucial for proving whether a function is a valid kernel.
   - The Kernel Trick - Medium – A clear explanation of the kernel trick and how kernel functions like $(1 + xy)^n$ work.
   - What is a Kernel Function in Machine Learning? – This article explains kernel functions, including polynomial kernels, and how to check if a function is a valid kernel.
2. **Counterexample for $k(x, y) = xy - 1$ Being Not a Kernel:**
   - Positive Semi-Definite Kernels – This page explains positive semi-definiteness and includes some examples of kernel matrices, which you can use to find counterexamples.
   - Counterexample of a Non-Kernel Function – StackExchange discussion on how to prove that certain functions (like $xy - 1$) are not positive semi-definite.

Part 2: Kernel SVM for Classification

1. **Hinge Loss Approximation:**
   - Hinge Loss and its Role in SVM – This article explains the concept of Hinge loss and its advantages over the 0-1 loss, including its use in SVMs.
   - Hinge Loss vs 0-1 Loss – StackExchange discussion about why hinge loss is used and how it approximates 0-1 loss.
2. **Lagrangian of the Soft SVM Problem:**
   - Dual SVM Problem – This article explains the derivation of the dual problem in SVMs, which is helpful for formulating the Lagrangian and working through the dual problem.
   - Soft Margin SVM Optimization – An overview of the Soft Margin SVM, including the Lagrangian formulation and how constraints are handled.
3. **Dual Problem with Kernel:**
   - Understanding the Dual Formulation of SVMs – This tutorial shows how to derive the dual form of the SVM using the kernel trick and how the kernel replaces the feature map.
   - The Kernel Trick and Dual Problem in SVM – This article walks through the derivation of the dual problem for SVMs with kernels.
4. **Prediction Function Using Kernel:**
   - Kernel SVM Prediction – This article explains how the kernel is used to express the prediction function in an SVM, even without directly using the feature map ϕ.
   - SVM Prediction Using Kernels – Explains the role of the kernel in prediction and how to express predictions without explicitly using the feature map.

These websites should provide a strong foundation for tackling your task without referencing any books. Let me know if you need any further assistance!

My second Prompt

Here I had to correct it since it gave me books and I would instead want Webpages:

I also have this task:

"

2 Exponential Families (1pts)Problem 3(1pts).

We consider the exponential familyp(x|η)=h(x) exp(ηTT(x)−A(η)),
(2.1)whereT(x) is asufficientstatistic,A(η)=logRh(x)eηTT(x)dxis
thepartitionfunction. Verify that∂∂ηiA(η)=Ep(x|η)[Ti(x)],
(2.2)∂2∂ηi∂ηjA(η)=Covp(x|η)[Ti(x),Tj(x)],(2.3)where Cov is the
covariance andTi(x) is the i-th component ofT(x)

"

Can you find some sources for me to read up on here too

Second Response

For the task on exponential families, the key concepts revolve around the **exponential family** of distributions, **sufficient statistics**, **partition functions**, and their derivatives. Below are some online sources that explain these concepts and can help you understand and verify the equations provided in your task.

1. Exponential Family of Distributions

- Exponential Family of Distributions (Wikipedia) – A good starting point to understand the general form of distributions in the exponential family, the components $h(x)$, $T(x)$, and the partition function $A(\eta)$.
- Understanding the Exponential Family – This section of the *Python Data Science Handbook* explains the exponential family, sufficient statistics, and how these concepts apply in probabilistic modeling and machine learning.

2. Sufficient Statistics and Derivatives

- Sufficient Statistics and the Exponential Family (UCLA) – A detailed PDF on the exponential family that goes through sufficient statistics and how they are derived from the likelihood function.
- Sufficient Statistics and Maximum Likelihood Estimation – This webpage explains the role of sufficient statistics in likelihood estimation and provides insights into the exponential family.

3. Partition Function and Its Derivatives

- Partition Function in Statistics (Wikipedia) – This article discusses the partition function $A(\eta)$ in the context of probability distributions, and how it normalizes the distribution.
- Exponential Family, Derivatives, and Expectation – This section touches on the derivative of the partition function and how the expectation of the sufficient statistic is related to the derivative.

4. Derivatives of the Partition Function

- Statistical Inference and Derivatives – This article focuses on the first and second derivatives of the log-partition function and connects them with expectations and covariances of sufficient statistics.
- The Log-Partition Function Derivative in Exponential Families – This lecture note explains the relationship between the partition function, its derivatives, and the expected values and covariances of sufficient statistics.

5. Covariance of Sufficient Statistics

- Covariance and Variance of Sufficient Statistics – Explains how the covariance of the sufficient statistics is related to the second derivative of the log-partition function.

These sources should help you understand the exponential family and verify the equations for the derivatives of the partition function as you work through the task. If you need more specific details, feel free to ask!

Task 3

My prompt

I have this final task, I've checked all my lecture papers and find nothing similar to this task. So I actually need these sources quite desperatly, so please find some good ones:
"
3 Maximum Likelihood Estimators (3pts)Problem 4(3pts).We consider the maximum likelihood estimation of the multivariate Gaussian distribution andits convergence properties. Recall that the density function of thed-dimensionalmultivariateGaussiandistributionwith meanµand covarianceΣisp(x|µ,Σ)=1(2π)d/2|Σ|
1/2exp−12(x−µ)⊤Σ−1(x−µ)!.Given i.i.d. samplesx1,x2,···,xNfromp(x|µ,Σ), whereµandΣare unknown parameters.1. Find the maximum likelihood estimators (MLE)ˆµMLandˆΣML.[Hint: Note that the multivariate Gaussiandistribution belongs to the exponential family and use some results in slides. You can also use the followingfact: Ifˆθis the MLE ofθ, then for any functionτ(θ),τ(ˆθ)is the MLE ofτ(θ).]2. ComputeE[ˆµML] andE[ˆΣML], where both expectations are taken with respect top(x1,x2,···,xN|µ,Σ).Are these estimators unbiased ?[Hint: estimatorˆµis unbiased ifE[ˆµ]=µ.]3. Show thatEh∥ˆµML−µ∥2i=TrΣN,(3.1)where TrΣis the trace of the matrixΣ.
"

Second Response

For your task on **Maximum Likelihood Estimators (MLE)** of the **multivariate Gaussian distribution** and their convergence properties, the following resources will help you understand the concepts and steps involved:

1. Maximum Likelihood Estimation (MLE) of Multivariate Gaussian

- Maximum Likelihood Estimation for Gaussian Distributions (Wikipedia) – This page covers the basic principles of MLE and includes the multivariate Gaussian case, explaining how to derive the MLE for the mean and covariance.
- Multivariate Normal Distribution and MLE – A StackExchange discussion showing how to derive the MLE for the multivariate normal distribution.
- MLE for Multivariate Gaussian – This article explains how to find the MLE for both the mean μ and the covariance Σ of a multivariate Gaussian, as well as some properties of the MLEs.

2. Exponential Family and MLE

- The Exponential Family of Distributions and MLE – Lecture notes that detail how the multivariate Gaussian fits into the exponential family and how to compute MLEs based on that.
- Exponential Family and MLE Derivations – This resource covers how exponential families relate to MLE and gives examples of finding MLEs in these families.

3. Unbiasedness of MLEs

- Unbiased Estimators in Statistics – A good source to understand what it means for an estimator to be unbiased and how to compute expected values of MLEs to verify unbiasedness.
- Expected Values of MLE for Multivariate Normal – Lecture notes on calculating the expected values of $\hat{\mu}_{MLE}$ and $\hat{\Sigma}_{MLE}$, helping you understand if they are unbiased.

4. Trace of Covariance and Its Role in the MLE

- The Trace Operator and Its Properties – This article explains the trace operator and its properties, which are useful when deriving results like $\mathbb{E}[\|\hat{\mu}_{ML} - \mu\|^2] = \mathrm{Tr}(\Sigma)/N$.
- Trace of Covariance in Multivariate Gaussian – A StackExchange discussion that walks through the derivation of the mean squared error for MLE estimators in the multivariate Gaussian case, including the trace of the covariance matrix.

5. Convergence and Asymptotics of MLE

- Asymptotic Properties of MLE – This page discusses the asymptotic properties of MLE, such as consistency and asymptotic normality, which can help you understand the behavior of the MLE as N increases.
- Asymptotic Normality of MLE – Explains the asymptotic distribution of MLEs and gives an introduction to their convergence properties.

6. General MLE Derivation Steps and Examples

- Deriving MLE for Multivariate Gaussian (Video) – This video walks through the derivation of MLE for the mean and covariance matrix of a multivariate Gaussian, which will be helpful for solving the first part of your task.
- Multivariate Gaussian and MLE in R – A StackExchange thread that provides a practical example of implementing MLE for multivariate normal distributions, which might help in understanding the concepts computationally.

These sources should give you the necessary tools to understand the MLE for the multivariate Gaussian distribution, including the derivation, unbiasedness, and convergence properties.