Third assignment in Machine learning 1 – 2023 – Paper 1

1 Principal component analysis (Deadline: 18-th October)

Suppose we have a dataset of N D-dimensional vectors $\{\mathbf{x}_n\}_{n=1}^N$, $\mathbf{x}_n \in \mathbb{R}^D$. We can write the entire dataset as an $N \times D$ matrix $\mathbf{X}$ (row n is $\mathbf{x}_n^\top$). We may wish to perform PCA on this data in the original data space, or in kernel space using kernel-PCA. In the latter case, the data are projected into feature-space, such that $\boldsymbol{\phi}_n = \boldsymbol{\phi}(\mathbf{x}_n)$ is an M-dimensional feature space representation of $\mathbf{x}_n$. Consider the procedure for PCA (which can be generalized to kernel-PCA):

- Step 1: Center $\mathbf{X}$ using its sample mean, producing a centered data matrix $\hat{\mathbf{X}}$.
- Step 2: Compute sample covariance $\mathbf{S}$ of the centered dataset.
- Step 3: Solve the eigenvalue problem as to find the decomposition $\mathbf{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$, where $\mathbf{U}$ is a square matrix whose k-th column is the eigenvector $\mathbf{u}_k$ of $\mathbf{S}$, and $\boldsymbol{\Lambda}$ is a diagonal matrix whose diagonal elements are the corresponding eigenvalues λ_k, *i.e.* $\Lambda_{kk} = \lambda_k$ and zero otherwise.

  The covariance matrix thus decomposes into a set of orthonormal (linearly independent) basis vectors and their corresponding eigenvalues. This allows us to order our new basis vectors into independent projections that preserve the largest variance of the data.
- Step 4: Pick the K eigenvectors with largest eigenvalues $\{\mathbf{u}_1, \dots, \mathbf{u}_K\}$.

  These will correspond to the projections that preserve the largest variance in the data. Since these projections are now linearly independent, the eigenvectors with smaller eigenvalues will only account for smaller variations in the data, and hence we can discard them without the original data being altered too much.
- Step 5: Project data onto the new K-dimensional basis.

Answer the following questions:

(a) Provide an expression for $\hat{\mathbf{x}}_n$ (with $\hat{\mathbf{x}}_n$ being a row vector from $\hat{\mathbf{X}}$). [.25 point]

(b) Prove that the average of $\hat{\mathbf{x}}_n$ (over N data vectors) is the $\mathbf{0}$ vector. [.25 point]

(c) Provide an expression for $\mathbf{S}$ in terms of $\hat{\mathbf{X}}$. [.25 point]

(d) What is the dimensionality of $\mathbf{S}$? [.25 point]

(e) What is the expression for the linear projection $\mathbf{L}$ that maps data vectors $\hat{\mathbf{x}}$ onto a K-dimensional sub-space, $\mathbf{y}_n = \mathbf{L}\hat{\mathbf{x}}_n$, such that it has zero mean and identity covariance.

(*i*) Write down the expression for **L**. [.25 point]

(*ii*) Prove that the average over N of y_n is 0. [.25 point]

(*iii*) Prove that the covariance of y_n is the identity. [.25 point]

(*iv*) What is the operator **L** called? [.25 point]

(*f*) Given a projection **L** that maps data vectors onto a K-dimensional subspace:

(*i*) How does varying K affect the resulting variance, information loss, and computation? [.75 point]

(*ii*) What trade-offs need to be considered when selecting K compared to D? [.25 point]

(*g*) PCA provides a linear projection from a large M-dimensional feature space to a lower dimensional feature space D.

- Can you propose a non-linear adaption of PCA which can learn non-linear projections (without using pre-defined kernels)? What would this method look like? [.5 point]

- Give one pro and one con of this approach in comparison with PCA. [.5 point]

(*h*) Which two steps (1→5) are the slowest in PCA and what is the computational complexity of each of these? And for the whole algorithm? Here, suppose K=D. What is a possible problem you observe when inspecting this complexity and how could this problem be overcome? [1 point]

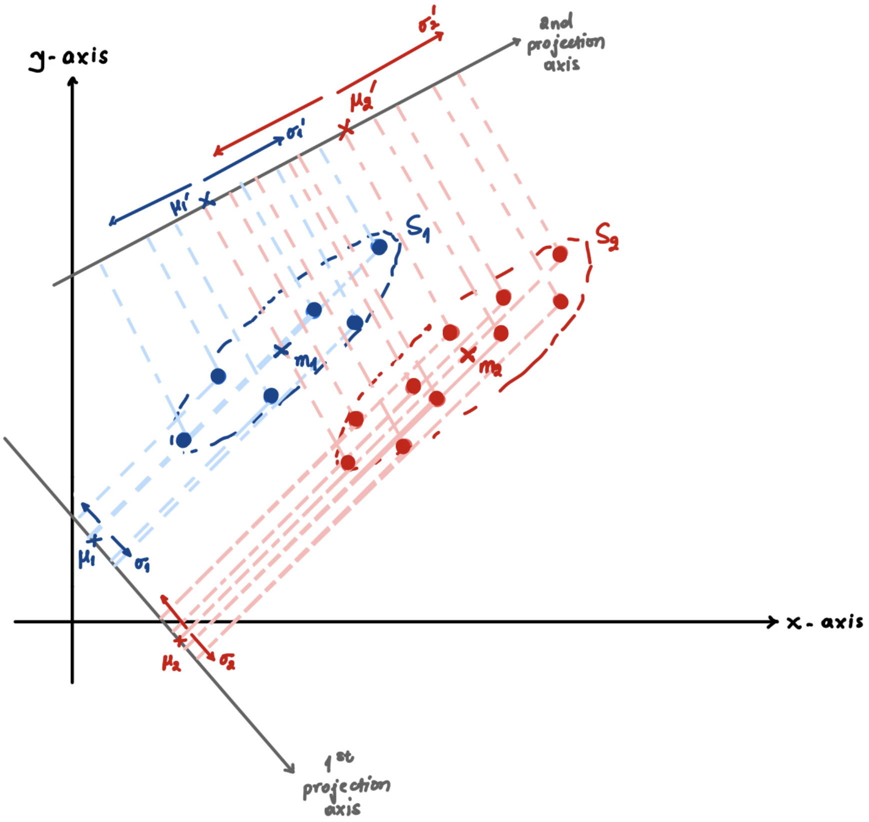
(*i*) PCA is an unsupervised method that does not take into account the given class information from your datasets. Could you name a dimensionality reduction method that can take into account the supervised information? [1 point]

Hint: Think about approaches covered during the classification lectures.

(*j*) Given the annotated dataset (see Figure), which projection do you want to calculate to better separate the data? How would you calculate this projection? [1 point]

In this figure, the data points with blue denote the first class, while red denotes the second class. The entities $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ denote the mean value for the initial space, while μ_1 and μ_2 or $\mu_1^{'}$ and $\mu_2^{'}$ are the means in the projected spaces.

Similarly, you can find the variance for each class in the figure s_1 and s_2.



(k) Can you formulate this projection in terms of $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$? Note: you do not need to perform any decomposition. [1 point]

Third assignment in Machine learning 1 – 2023 – Paper 1

2 Probabilistic PCA - A general latent space distribution (Deadline: 18-th October)

Principal Component Analysis (PCA) is an often used technique for dimensionality reduction. In this approach, we linearly project data onto the subspace of lower dimensionality. However, this approach can be extended to be more general by formulating a latent variable model named probabilistic PCA. In this approach, the PCA can be expressed as the maximum likelihood solution probabilistic PCA. There are multiple advantages of the probabilistic PCA over the conventional PCA. To name a few, we can associate a likelihood function to the probabilistic PCA which allows a direct comparison with other probabilistic density models, probabilistic PCA can be used to model class-conditional densities and can thus be used in classification problems, and also we can run the model generatively to provide samples from the modeled distribution.

Probabilistic PCA is an example of the linear-Gaussian framework, where both marginal and conditional distributions are Gaussian. We first define a latent variable $\mathbf{z}$, corresponding to the principal-component subspace. We can then define a prior distribution $p(\mathbf{z})$ over the latent variable $\mathbf{z}$, and also the conditional distribution $p(\mathbf{x}|\mathbf{z})$, which is given by

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})\,.$$

The prior distribution over $\mathbf{z}$ is usually given by a zero-mean unit-covariance Gaussian

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})\,.$$

In this case, the marginal distribution $p(\mathbf{x})$ is also a Gaussian, and is given by

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I})\,.$$

However, suppose we replace the zero-mean and the unit-covariance latent space distribution by a general Gaussian distribution of the form

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{m}, \boldsymbol{\Sigma}).$$

In this problem, we wish to show that by redefining parameters of the model, this assumption on the prior leads to an identical model for the marginal distribution $p(\mathbf{x})$ over the observed variables for any valid choice of $\mathbf{m}$ and $\boldsymbol{\Sigma}$. We will derive this result in multiple steps.

(*a*) We can express the random variable $\mathbf{z}$ as $\mathbf{z} = \mathbf{m} + \boldsymbol{\epsilon}_z$, with noise $\boldsymbol{\epsilon}_z \sim \mathcal{N}(0, \boldsymbol{\Sigma})$. Write out a similar expression for the variable $\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}_x$, and explain what distribution variable $\mathbf{x}$ follows. From which distribution is $\boldsymbol{\epsilon}_x$ sampled from? [1 point]

(*b*) Find the expectation value of the variable $\mathbf{x}$ using the linearity property of the expected value. [1 point]

(c) Find the covariance of the variable $\mathbf{x}$ using the definition of the covariance $\mathrm{cov}[\mathbf{x}, \mathbf{x}]$. [2 point]
Hint: The following identity might be helpful: $\mathrm{Var}[\mathbf{AY}] = \mathbf{A}\mathrm{Var}[\mathbf{Y}]\mathbf{A}^{\mathrm{T}}$, where $\mathbf{A}$ is a matrix, and $\mathbf{Y}$ is a random variable.

(d) To show that using the general Gaussian prior still leads to an identical model $p(\mathbf{x})$, we have to be able to write the distribution $p(\mathbf{x})$ in the form $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{W}}\tilde{\mathbf{W}}^{\mathrm{T}} + \sigma^2\mathbf{I})$. Find the appropriate expressions for $\tilde{\boldsymbol{\mu}}$ and $\tilde{\mathbf{W}}$. [1 point]

Third assignment in Machine learning 1 – 2023 – Paper 1

3 Mixtures of experts (Deadline: 18-th October)

In class, you discussed and were introduced to mixture models as a way to perform unsupervised learning tasks, *e.g.* clustering. Mixture models can be similarly used for supervised learning tasks. In this question, we will discuss and explore the Mixtures of Experts (MoEs), a model that softly partitions the input space and learns a supervised model for each area.

Consider that you have K expert models available in order to model a specific dataset of N data points $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_n \in \mathbb{R}^D$ and y_n corresponds to the ground truth label for $\mathbf{x}_n$. Let $\mathbf{z}_n \in \mathbb{R}^K$ correspond to a one-hot encoding of a categorical random variable for data point $\mathbf{x}_n$ that denotes which of the K expert models is 'active'. If expert model k is active for datapoint $\mathbf{x}_n$, it means that model k provides the prediction for datapoint $\mathbf{x}_n$. The strength of an MoE approach is that it trains expert models that each specialise in a specific portion of the data.

Then, let $\boldsymbol{\Theta}$ be a matrix in $\mathbb{R}^{D\times K}$ that contains the D-dimensional column vector of parameters for each expert. We will assume that each y_i is a continuous random variable at the $[0, \infty)$ interval and is distributed according to an exponential distribution with a rate $\lambda > 0$. Given the aforementioned assumptions, each expert $k \in K$ has the following linear predictive model:

$$p(y_n|\mathbf{x}_n, \mathbf{z}_n, \boldsymbol{\Theta}) = p(y_n|\mathbf{x}_n, \boldsymbol{\theta}_k = \boldsymbol{\Theta}\mathbf{z}_n) = \text{Exponential}\big(y_n|\lambda = \exp\big(\boldsymbol{\theta}_k^T\mathbf{x}_n\big)\big)$$

where $\mathbf{z}_n$ is the one-hot-encoded vector representation of the categorical variable $\mathbf{z}_n$ and

$$\text{Exponential}(y|\lambda) = \lambda \exp(-\lambda y) \text{ for } y \geq 0.$$

The flexibility of MoEs stems from the fact that there is a "routing" mechanism which determines how relevant each of the K experts is for a specific datapoint $\mathbf{x}_n$. As in this case we have a discrete set of K experts, a simple linear routing mechanism is the following:

$$p(z_{nk} = 1|\mathbf{x}_n, \boldsymbol{\Phi}) = \pi_{nk} = \frac{\exp(\boldsymbol{\phi}_k^T\mathbf{x}_n)}{\sum_j \exp(\boldsymbol{\phi}_j^T\mathbf{x}_n)}$$

where $\boldsymbol{\Phi}$ is a matrix in $\mathbb{R}^{D\times K}$ that contains all of the parameters of the routing function, i.e. $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_K]$.

(a) Note that the output of the routing function falls between 0 and 1. Thus, we need to construct our vector $\mathbf{z}_n$ based on these values.
Write down the formula to decide each element z_{nk} of $\mathbf{z}_n$, assuming you want the most relevant expert to be active for datapoint $\mathbf{x}_n$. Hint: remember that $\mathbf{z}_n$ is one-hot encoded. [0.5 point]

As a-priori we have no information about which of the experts is responsible for generating a particular prediction, we have to marginalize over all possible experts in order to compute the likelihood of an observed point. With this information answer the following questions:

(b) Write down the likelihood of the entire dataset, $p(\mathbf{y}|\mathbf{X}, \mathbf{\Theta}, \mathbf{\Phi})$, and take its log under the i.i.d. assumption. [1 point]

(c) Write down the posterior probability r_{ni} of expert i producing the label y for datapoint n. We will also refer to this as the responsibility of expert i for datapoint n. [1 point]

(d) Take the derivative of the log-likelihood w.r.t. the parameters of each expert $\boldsymbol{\theta}_i$ and the parameters of the routing mechanism for each expert $\boldsymbol{\phi}_i$. Do not substitute expressions for the probabilities but rather provide your answer in terms of $p(y_n|\mathbf{x}_n, z_n, \boldsymbol{\theta}_i)$, $p(z_{nk} = 1|\mathbf{x}_n, \mathbf{\Phi})$. Make sure to express the derivatives in terms of the responsibilities of each expert r_{ni}. (Hint: $\frac{\partial f(x)}{\partial x} = f(x)\frac{\partial \log f(x)}{\partial x}$), as that term will be present in the derivatives for both $\boldsymbol{\theta}_i, \boldsymbol{\phi}_i$. [1.5 points]

(e) Now insert the explicit expression for each of the respective probability distributions and compute the final derivatives for $\boldsymbol{\theta}_i, \boldsymbol{\phi}_i$. [1.5 points]

(f) Write down an iterative algorithm that maximizes the log-probability of the data by jointly optimizing the $\mathbf{\Theta}$ and $\mathbf{\Phi}$ parameters. Make use of appropriate convergence criteria. [1 point]

(g) For this question assume that instead of having $\mathbf{z}_n$ as a one-hot encoding, we have $z_{nk} = \pi_{nk}$. Thus, each element of $\mathbf{z}_n$ falls between 0 and 1. To compute our final prediction $\hat{y_n}$ for datapoint $\mathbf{x}_n$, we will now weigh the prediction of each expert by its relevancy. Write down the formula for determining the final prediction $\hat{y_n}$. Denote the prediction of expert k with $\hat{y_{nk}}$. [0.5 point]