4 Examiner's comment:

The least popular question, but well handled by most.

Consider the k-means clustering algorithm which seeks to minimise the cost function

$$C = \sum_{n=1}^{N} \sum_{k=1}^{K} s_{nk} \|x_n - m_k\|^2$$

where m_k is the mean (centre) of cluster k, x_n is data point n, $s_{nk} = 1$ signifies that data point n is assigned to cluster k, and there are N data points and K clusters.

(a) Given all the cluster assignments s_{nk} (with the constraint that each data point must be assigned to one cluster, that is, $\sum_k s_{nk} = 1$ for all n, and $s_{nk} \in \{0,1\}$ for all n and k), derive the value of the means $\{m_k\}$ which minimise the cost C and give an interpretation in terms of the k-means algorithm. [30%]

(b) Give an interpretation of the k-means algorithm in terms of a probabilistic model. Describe up to three generalisations based on this probabilistic model. [40%]

(c) You are applying the k-means algorithm to a large collection of images, where most of the images are not labelled, but you have labels for a few of the images (e.g. "cat", "dog", "person", "car"). You would like to modify your k-means algorithm so that images with the same label are always in the same cluster, and images with different labels are never in the same cluster. Describe a modified version of the algorithm that would do this. [30%]

SOLUTION

(a) Given the cluster assignments, the problem decomposes into separate minimisations over each mean k, that is, $C = \sum_k C_k$. Since the cluster assignments are binary, for each mean we have a cost function

$$C_k = \sum_{n=1}^{N} s_{nk} \|x_n - m_k\|^2 = \sum_{n:s_{nk}=1} \|x_n - m_k\|^2$$

Minimising over m_k results in

$$m_k = \frac{\sum_{n:s_{nk}=1} x_n}{\sum_n s_{nk}}$$

which is simply the Euclidean mean of the data points assigned to cluster k.

(b) The k-means algorithm is closely related to the Gaussian mixture model, a probabilistic model for density esimation. In fact, the k-means cost is equal up to a constant to the (negative) log likelihood of a Gaussian mixture model under the

following assumptions: (1) the Gaussians have means m_k and covariances that are a multiple of the identity matrix, (2) the Gaussians all have equal mixing proportions, (3) the assignment variables which are actually hidden are treated as parameters and optimised rather than summed out. Upto these three constraints, k-means is almost identical to the EM algorithm. Once this relationship is established several generalisations become possible: (1) using different covariance matrices for each cluster to allow for elongated clusters at different orientations, (2) allowing different mixing proportions so that some clusters can be bigger than others, (3) handling partial membership of data points in clusters by accounting for the uncertainty in the assignment variables s_{nk}, (4) use of models other than the Gaussian to capture each cluster (e.g. mixtures of any other distribution), and (5) Bayesian generalisations whereby the number of clusters can be learned from data, and the uncertaintly in clustering is represented in the inference.

(c) Assume that the number of clusters K is equal or greater than the number of labels (otherwise the constraints can't be satisfied). Initalise assignments so that the labelled images belong to separate clusters (e.g. all "cats" in cluster 1, all "dogs" in cluster 2, etc). Run the k-means algorithm as before on all the unlabelled data, but ensure that the assignments for the labelled data remain unchanged. Since the constraints are imposed at initialisation and kept at each iteration, this will converge to a solution which is a (local) minimum of the cost C subject to the imposed constraints.

END OF PAPER

4 **ASSESSOR's comment:**

This year question 4 was popular, which is a pleasing shift compared to previous exams. a) Was handled well by most. b) Caused some confusion, in that many tried set the joint likelihood to be > 0.5, which is a harder problem, whereas the question requires the each of the two data points be classified with prob. > 0.5, a much easier problem. This error then made it difficult to find the parameter regions and the maximum likelihood value. d) Most had a vague idea of how Bayesian methods modifiy the situation, though few had enough knowledge to make the full 20

Consider a binary classification problem where the data $\mathscr{D}$ consists of N data points, $\mathscr{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, x_n is a real scalar and $y_n \in \{0, 1\}$, and the goal is to predict class labels y for new x.

Assume a very simple logistic classification model in which the class labels were produced independently and identically from the following model:

$$P(y_n = 1 | x_n, a, b) = \sigma(ax_n + b)$$

where σ is the logistic function, $\sigma(z) = \frac{1}{1+\exp(-z)}$, and a and b are the parameters of the classifier.

(a) Write down the likelihood of a and b for the data $\mathscr{D}$ and describe an algorithm to optimise this likelihood as a function of a and b. [40%]

(b) Consider a data set consisting of only two data points, $\mathscr{D} = \{(-2, 0), (3, 1)\}$. For this data set, describe the set of parameters which classify both data points correctly with probability greater than 0.5. Furthermore, what is the maximum achievable likelihood and describe the set of parameters achieving this maximum. [40%]

(c) Explain how Bayesian learning of the parameters might give more reasonable inferences about a and b from the data set in part (b) than maximum likelihood (ML) and how the Bayesian predictions about future labels differ from the ML predictions. [20%]

SOLUTION:

(a) The likelihood is

$$p(y_1,\ldots,y_N|x_1,\ldots,x_N,a,b) = \prod_{n=1}^{N} p(y_n|x_n,a,b) \quad (1)$$

$$= \prod_{n=1}^{N} \sigma(ax_n+b)^{y_n}(1-\sigma(ax_n+b))^{1-y_n} \quad (2)$$

An algorithm to optimise this is steepest gradient ascent in the likelihood which takes derivatives of the log likelihood and moves parameters in the direction of these derivatives (Pattern Processing Lecture 3, slide 8).

[Optional but not required for full marks: the steps for the batch version of this algorithm with step size *eta* are:

$$a^{[t+1]} = a^{[t]} + \eta \sum_n (y_n - \sigma(a^{[t]}x_n + b^{[t]}))x_n \quad (3)$$

$$b^{[t+1]} = b^{[t]} + \eta \sum_n (y_n - \sigma(a^{[t]}x_n + b^{[t]})) \quad (4)$$

]

(b) To classify both data points correctly (with prob. > 0.5) we require that

$$-2a+b < 0 \quad (5)$$

$$3a+b > 0 \quad (6)$$

Negating the first inequality and adding the two we find that

$$a > 0$$

and subsequently solving for b we get that

$$-3a < b < 2a.$$

The region of (a,b) parameter space defined by these two ineqalities gives us correct classification. The larger the value of a (for b satisfying the above constraint), the higher the likelihood, since the σ function increases monotonically to 1. In the limit of $a \to \infty$ the likelihood is 1. So the maximum achievable likelihood is 1, and this occurs when $a \to \infty$ as long as b stays in $-3a < b < 2a$.

(c) In part (b), because the data is linearly separable, the ML parameters can go to infinity, resulting in a very sharp and confident classification boundary. If you put a prior on (a,b) the posterior will put more mass in the same region defined as in part (b), but still reflect a reasonable amount of uncertainty about what a and b should be. By averaging over the posterior we get predictions that are not very confident; this makes sense, since we've only observed two data points.

Please set a version number using \version

4 SOLUTION

(a) The likelihood is

$$p(\mathbf{y}|\mathbf{x},a,\sigma^2) = \prod_n p(y_n|x_n,a,\sigma^2) = \prod_n \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\{-\frac{1}{2\sigma^2}(y_n - ax_n)^2\}$$

Equivalently we maximise

$$\log p(\mathbf{y}|\mathbf{x},a,\sigma^2) = \sum_n -\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y_n - ax_n)^2$$

Taking derivatives w.r.t. a and setting to zero gives

$$\sum_n \frac{1}{2\sigma^2}(y_n - ax_n)x_n = 0$$

Solving for a gives

$$a_{\mathrm{ML}} = \left(\sum_{n=1}^N y_n x_n\right) / \left(\sum_{n=1}^N x_n^2\right)$$

which is the scalar version of the "normal equations". Taking derivatives w.r.t. σ^2 and setting to zero we get:

$$\begin{aligned} -\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_n (y_n - ax_n)^2 &= 0 \\ N\sigma^2 &= \sum_n (y_n - ax_n)^2 \\ \sigma^2_{\mathrm{ML}} &= \frac{1}{N}\sum_{n=1}^N (y_n - ax_n)^2 \end{aligned}$$

which is the average of the residual errors.

(b) The maximum of the likelihood for this model will be higher than for the model in part (a) since this more complex model has more parameters to fit the data. The value of σ^2 will be lower than for part (a) since again the model can fit the data better and so the average residuals will be smaller (see solution for σ^2_{ML} in part (a)).

(c) Laplacian noise has heavier tails than Gaussian noise, so in general using such a model might be a good idea if we expect to have *outliers*—points where the actual y values are far from the linear prediction. Because this model can better handle outliers it is considered more *robust* than the model in part (a).

END OF PAPER

SOLUTION

(a) Given the means, the problem decomposes into separate minimisations over each data point n. For data point n, the solution is to set $s_{nk} = 1$ for the value k which has the smallest distance $\|x_n - m_k\|^2$, and to set $s_{nk'} = 0$ for all $k' \neq k$. In terms of the k-means algorithm, the interpretation is that we assign each data point to the cluster with the nearest centre, as measured by Euclidean distance.

(b) Minimising C as a function of K is *not* a good idea. There are (at least) two ways to see this. One is that the optimal k-means cost for K could always be decreased by adding a new $K+1$st centre, since the optimal solution for K is generally a suboptimal case for $K+1$. A second way is to consider the extreme where we have as many clusters as data points, $K = N$. Then clearly we can obtain a cost $C = 0$ simply by placing each mean on a distinct data point (e.g. $m_n = x_n$). Since $C \geq 0$, this is the lowest possible cost solution no matter how the data is distributed, so it gives no insight into the actual number of clusters in the data.

(c) Running PCA dimensionality reduction on the data, and then k-means, will not in general result in the same solution as running k-means directly on the high dimensional data. To see this, consider in general that $y_n = Wx_n$ is the lower dimensional PCA projection of x_n. Running k-means on $\{y_n\}$ means that we are mininising $\tilde{C} = \sum_{n,k} s_{nk} \|Wx_n - \tilde{m}_k\|^2$ rather than $C = \sum_{n,k} s_{nk} \|x_n - m_k\|^2$. Assume we initialise all $\tilde{m}_k = Wm_k$ (the low dimensional projection of the means). Then one step of k-means for $\tilde{C}$ assigns $s_{nk} = 1$ if $\|Wx_n - Wm_k\|^2 = (x_n - m_k)^\top W^\top W(x_n - m_k)$ is minimised, rather than $\|x_n - m_k\|^2$. Therefore the PCA k-means corresponds to using a non-Euclidean norm to find the nearest mean, and generally only coincides with the original k-means when $W^\top W = I$.

END OF PAPER

4 Consider a dataset of observations $\mathscr{D} = \{(\mathbf{x}_n, y_n)\}$ where $n = 1, \ldots, N$, and N is the total number of data points. $\mathbf{x}_n$ is a two dimensional vector. A regression model of the following form is to be trained using the following form of regression

$$y_n = \mathbf{a}^{\mathrm{T}}\mathbf{x}_n + \varepsilon_n$$

where ε is independent zero-mean Gaussian noise with variance σ^2.

(a) Write down the log-likelihood $\log(p(y_1, \ldots, y_N | \mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{a}, \sigma^2))$ in terms of $y_1, \ldots, y_N, \mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{a}, \sigma^2$. [20%]

(b) Show that the maximum likelihood estimate of the regression parameters, $\hat{\mathbf{a}}$, can be expressed in the following form

$$\hat{\mathbf{a}} = \mathbf{C}^{-1}\mathbf{B}$$

You should clearly state the forms of the two matrices $\mathbf{C}$ and $\mathbf{B}$. The following equality may be useful

$$\frac{\partial \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = \mathbf{x}^{\mathrm{T}}\mathbf{A} + \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$$

for any square matrix $\mathbf{A}$ and vector $\mathbf{x}$. [50%]

(c) A non-linear transformation $\phi(\mathbf{x}_n)$ is applied to the observations $\mathbf{x}_n$. The size of the resulting vector $\phi(\mathbf{x}_n)$ is d. Regression based on these transformed data points is then performed. Now

$$y_n = \mathbf{a}^{\mathrm{T}}\phi(\mathbf{x}_n) + \varepsilon_n$$

where ε is again independent zero-mean Gaussian noise with variance σ^2. Briefly discuss how the performance of the regression process and the estimation of the regression parameters may be impacted as the size of the transformed features, d, increases. [30%]

Solution:

This question is fully covered in lectures.

(a) The log-likelihood can be written as

$$\begin{aligned}\mathscr{L}(\mathbf{a}) &= \log(p(y_1,\ldots,y_N|\mathbf{x}_1,\ldots,\mathbf{x}_N,\mathbf{a},\sigma^2)) \\ &= \sum_{n=1}^{N}\log\left(\mathscr{N}(y_n-\mathbf{a}^{\mathrm{T}}\mathbf{x}_n;0,\sigma^2)\right) \\ &= \sum_{n=1}^{N}\left(-\frac{1}{2}\log(\sqrt{2\pi\sigma^2})-\frac{1}{2\sigma^2}\left[y_n^2-2y_n\mathbf{x}^{\mathrm{T}}\mathbf{a}+\mathbf{a}^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{a}\right]\right)\end{aligned}$$

(b) Rearranging the above expression

$$\mathscr{L}(\mathbf{a}) = \frac{N}{2}\log(\sqrt{2\pi\sigma^2})-\frac{1}{2\sigma^2}\left(\left[\sum_{n=1}^{N}y_n^2\right]-2\left[\sum_{n=1}^{N}y_n\mathbf{x}^{\mathrm{T}}\right]\mathbf{a}+\mathbf{a}^{\mathrm{T}}\left[\sum_{n=1}^{N}\mathbf{x}\mathbf{x}^{\mathrm{T}}\right]\mathbf{a}\right)$$

Differentiate this expression with respect to $\mathbf{a}$

$$\frac{\partial\mathscr{L}(\mathbf{a})}{\partial\mathbf{a}} = -\frac{1}{2\sigma^2}\left(-2\left[\sum_{n=1}^{N}y_n\mathbf{x}\right]+2\left[\sum_{n=1}^{N}\mathbf{x}\mathbf{x}^{\mathrm{T}}\right]\mathbf{a}\right)$$

Equating this to zero and yields

$$\hat{\mathbf{a}} = \left(\sum_{n=1}^{N}\mathbf{x}\mathbf{x}^{\mathrm{T}}\right)^{-1}\left[\sum_{n=1}^{N}y_n\mathbf{x}\right]$$

Thus

$$\mathbf{C} = \sum_{n=1}^{N}\mathbf{x}\mathbf{x}^{\mathrm{T}};\quad \mathbf{D} = \sum_{n=1}^{N}y_n\mathbf{x}^{\mathrm{T}}$$

(c) Transforming the observation into a high-dimensional space $\phi(\mathbf{x}_n)$ should improve performance of modelling the training data as there are more parameters. Should mention:

•though better on the training data there may be issues generalising to unseen data

•the estimation requires an inversion of $\mathbf{C}$. As the dimensionality of the transformed data increases this may not be invertible due to the number of dimensions being greater than N, or numerical accuracy issues.

END OF PAPER

Version 1

2011

4 Examiner's comment:

Very unpopular, and poorly answered. As is often the case with this part of the course, this quesion was fairly easy marks for those who had learned the material.

Consider a binary classification problem with scalar real-valued observations x, and class labels $y \in \{0, 1\}$. Assume a model with parameters θ where

$$p(y=1|x,\theta) = \frac{1}{1+e^{-\theta x+\frac{1}{2}}}$$

(a) Describe the online learning rule for learning the parameter θ assuming the learning algorithm receives one data point at a time. [40%]

(b) Consider a data set $\mathscr{D}$ consisting of three data points: $(x_1 = 0,\ y_1 = 1)$, $(x_2 = -1,\ y_2 = 0)$, and $(x_3 = 1,\ y_3 = 1)$. Compute the likelihood for the parameters θ given this data set $\mathscr{D}$. [30%]

(c) Characterise the solution(s) to the maximum likelihood estimate of θ in part (b) above. Discuss properties of these solution(s), indicating any problems with the result and possible ways of resolving those problems. [30%]

Solution:

(a) This material was basically covered in lecture 3, slide 8. The idea is to write down the likelihood function for each data point and move θ a small amount in the direction of this gradient.

$$\begin{aligned}
P(y|x,\theta) &= \sigma(\theta x-0.5)^y(1-\sigma(\theta x-0.5))^{(1-y)} \\
\ln P(y|x,\theta) &= y\ln\sigma(\theta x-0.5)+(1-y)\ln(1-\sigma(\theta x-0.5)) \\
\text{use: } \frac{\partial \ln\sigma(z)}{\partial z} &= \sigma(-z) \\
\text{let: } z &= \theta x-0.5 \\
\frac{\partial \ln P(y|x,\theta)}{\partial \theta} &= yx\sigma(-z)-(1-y)x\sigma(z) \\
&= yx-yx\sigma(z)-x\sigma(z)+yx\sigma(z) \\
&= (y-\sigma(z))x
\end{aligned}$$

The online learning rule changes the parameters θ at each step t by a small step η in the

(cont.

direction given by the maximum likelihood step above:

$$\theta_{t+1} \leftarrow \theta_t + \eta(y_t - \sigma(\theta_t x_t - 1/2))x_t$$

(b) Multiplying the probabilities of each observation we get that the likelihood is:

$$\left(\frac{1}{1+e^{1/2}}\right)\left(\frac{e^{\theta+1/2}}{1+e^{\theta+1/2}}\right)\left(\frac{1}{1+e^{-\theta+1/2}}\right)$$

or equivalently:

$$\left(\frac{1}{1+e^{1/2}}\right)\left(\frac{1}{1+e^{-\theta-1/2}}\right)\left(\frac{1}{1+e^{-\theta+1/2}}\right)$$

(c) To maximise the above likelihood we write down the log likelihood and drop additive constants getting for the first expression above:

$$L = \theta - \ln(1+e^{\theta+1/2}) - \ln(1+e^{-\theta+1/2})$$

for the second expression:

$$L = -\ln(1+e^{-\theta-1/2}) - \ln(1+e^{-\theta+1/2})$$

The second expression in the solution to (b) is easier to analyse. It's clear that the likelihood increases monotonically as θ increases. This means that the maximum likelihood occurs at the limit of $\theta \to \infty$. Looking at the configuaration of the data, this solution makes sense since the data is perfectly separable and increasing θ simply increases the slope of the logistic function.

The solution is problematic because of the basis of three data points the model will make absolutely certain predictions for new points. Two ways of resolving this problem are either to add a penalty term or prior to the log likelihood function, or to do Bayesian learning of the parameter θ (see Lecture 3:Classification, Slide 10).

END OF PAPER

4 Consider a binary classification problem with scalar real-valued observations x, and class labels $y \in \{0, 1\}$. Assume that $p(x|y=0)$ is a Gaussian distribution with mean 0 and variance 2, and $p(x|y=1)$ is a Gaussian distribution with mean 1 and variance 2. Furthermore, assume that $p(y=0) = p(y=1) = 1/2$.

(a) Compute the probability that given an observation $x=2$, its corresponding class label is $y=1$. [30%]

(b) Derive the general expression for $p(y=0|x)$ as a function of x, and discuss how this relates to logistic classification. [40%]

Now assume that you fit a maximum likelihood Gaussian distribution $p(x|y=0)$ with mean μ_0 and variance σ_0^2 to the observed data with label $y=0$, and similarly you fit a separate maximum likelihood Gaussian distribution $p(x|y=1)$ with mean μ_1 and variance σ_1^2 to the observed data with label $y=1$.

(c) Describe several ways in which the above procedure differs from maximum likelihood logistic classification, paying particular attention to the role of the variances and likelihood that is being optimised. [30%]

ANSWERS

(a) By Bayes rule:

$$
\begin{aligned}
p(y=1|x=2) &= \frac{p(x=2|y=1)\,p(y=1)}{p(x=2|y=1)\,p(y=1) + p(x=2|y=0)\,p(y=0)} \\
&= \frac{\frac{1}{\sqrt{2\pi 4}}\exp\{-\frac{1}{2}\frac{(2-1)^2}{2^2}\}\cdot\frac{1}{2}}{\frac{1}{\sqrt{2\pi 4}}\exp\{-\frac{1}{2}\frac{(2-1)^2}{2^2}\}\cdot\frac{1}{2} + \frac{1}{\sqrt{2\pi 4}}\exp\{-\frac{1}{2}\frac{(2-0)^2}{2^2}\}\cdot\frac{1}{2}} \\
&= \frac{\exp\{-\frac{1}{2}\frac{(2-1)^2}{2^2}\}}{\exp\{-\frac{1}{2}\frac{(2-1)^2}{2^2}\} + \exp\{-\frac{1}{2}\frac{(2-0)^2}{2^2}\}} \\
&= \frac{\exp\{-\frac{1}{8}\}}{\exp\{-\frac{1}{8}\} + \exp\{-\frac{1}{2}\}} \\
&= \frac{1}{1+e^{-3/8}}
\end{aligned}
$$

(TURN OVER for continuation of Question 4

(b) Again by Bayes rule:

$$\begin{aligned} p(y=0|x) &= \frac{\exp\{-\frac{1}{2}\frac{x^2}{2^2}\}}{\exp\{-\frac{1}{2}\frac{x^2}{2^2}\}+\exp\{-\frac{1}{2}\frac{(x-1)^2}{2^2}\}} \\ &= \frac{1}{1+\exp\{\frac{1}{8}x^2-\frac{1}{8}x^2+\frac{x}{4}-\frac{1}{8}\}} \\ &= \frac{1}{1+\exp\{\frac{x}{4}-\frac{1}{8}\}} \end{aligned}$$

This is a logistic function, exactly as in logistic classification. So the classification probabilities are equivalent to that of logistic classication.

(cont.

2009

4. Consider the k-means clustering algorithm which seeks to minimise the cost function

$$C = \sum_{n=1}^{N} \sum_{k=1}^{K} s_{nk} \|x_n - m_k\|^2$$

where m_k is the mean (centre) of cluster k, x_n is data point n, $s_{nk} = 1$ signifies that data point n is assigned to cluster k, and there are N data points and K clusters.

(a) Given all the assignments $\{s_{nk}\}$, derive the value of m_k which minimises the cost C and give an interpretation in terms of the k-means algorithm.

Answer

Solve by taking derivatives and setting to zero.

$$\begin{aligned} \frac{\partial C}{\partial m_k} &= \sum_{n=1}^{N} s_{nk} \frac{\partial}{\partial m_k} (x_n - m_k)^\top (x_n - m_k) \\ &= \sum_{n=1}^{N} s_{nk}(-2x_n + 2m_k) = 0 \\ m_k &= \frac{\sum_{n=1}^{N} s_{nk} x_n}{\sum_{n=1}^{N} s_{nk}} \end{aligned}$$

This equation can be interpreted as follows: m_k is set to the mean of the data points assigned to cluster k.

(b) Give a probabilistic interpretation of k-means and describe how it can be generalised to unequal cluster sizes and non-spherical (elongated) clusters as shown in Fig. 1 below.

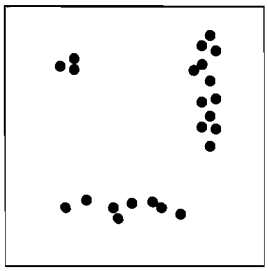

Figure 1:

Answer

K-means can be interpreted as an algorithm for fitting maximum likelihood parameters toa mixture of Gaussians where each Gaussian has spherically symmetric (i.e. isotropic) covariance matrix $\sigma^2 I$ and the Gaussians have equal proportions of data assigned to them $w_k = 1/K$ for all k (from lecture notes).

To generalise to unequal cluster sizes we allow w_k to vary, and to allow for elongated clusters we allow the covariance matrices for each Gaussian to vary and potentially be unequal.

(c) In many real-world applications, data points arrive sequentially and one wants to cluster them as they come in. Devise a sequential variant of the k-means algorithm which takes in one data point at a time and updates the means $\{m_1, \ldots, m_K\}$ sequentially without revisiting previous data points. Describe your sequential algorithm.

Answer

There are many possible answers, but here is one sequential variant of k-means:

- Assign the first K data points to the K clusters, and set $m_k = x_k$, and $n_k = 1$ (the number of points in cluster k).
- For each subsequent data point, x_n find the closest cluster centre, say m_k. Assign to this cluster and set:

$$m_k \leftarrow \frac{n_k}{n_k + 1} m_k + \frac{1}{n_k + 1} x_n$$

$$n_k \leftarrow n_k + 1$$

This algorithm has the property that m_k will always be the mean of all the data points assigned to it. One problem with this algorithm is that it is very senstitive to the first K points that arrive.

4 Consider a data set of pairs of observations $\mathscr{D} = \{(x_n, y_n)\}$ where $n = 1, \ldots, N$ and N is the total number of data points. Assume we wish to learn a regression model

$$y_n = ax_n + \varepsilon_n$$

where ε_n is independent zero-mean Gaussian noise with variance σ^2.

(a) Write down the log likelihood $\log p(y_1, \ldots, y_N | x_1, \ldots, x_N, a, \sigma^2)$ in terms of $y_1, \ldots, y_N, x_1, \ldots, x_N, a, \sigma^2$. [40%]

(b) Assume the following data set of $N = 4$ pairs of points

$$\mathscr{D} = \{(0,1), (1,2), (2,0), (3,4)\}$$

Solve for the maximum likelihood estimates of a and σ^2. [40%]

(c) Assume the same data set, but instead a regression model that predicts x given y:

$$x_n = by_n + \varepsilon_n$$

Is the maximum likehood estimate of b equal to $\frac{1}{a}$? Explain why or why not, giving a derivation if necessary. [40%]

Solution:

(a)

$$\begin{aligned} \log p(y_1, \ldots y_N | x_1, \ldots, x_N, a, \sigma^2) &= \sum_{n=1}^{N} \log p(y_n | x_n, a, \sigma^2) \\ &= -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - ax_n)^2 \end{aligned}$$

(b) Solving for a reduces to minimising

$$(2-a)^2 + (0-2a)^2 + (4-3a)^2$$

Taking derivatives

$$-2(2-a) - 4(0-2a) - 6(4-3a) = 0$$

$$-4+2a+8a-24+18a=0$$

therefore $a=1$. Computing the average squared residuals for σ^2.

$$\sigma^2=\frac{1}{4}[1+1+4+1]=\frac{7}{4}$$

(c) No the ML estimate of b is not $1/a$ since errors are being measured in x now. In fact, minimising $(0-b)^2+(1-2b)^2+(2-0b)^2+(3-4b)^2$ we get $42b=28$, so $b=2/3$.

END OF PAPER

4 Consider a data set of pairs of observations $\mathscr{D} = \{(x_n, y_n)\}$ where $n = 1, \ldots, N$ and N is the total number of data points. Assume we wish to learn a regression model

$$y_n = ax_n + \varepsilon_n$$

where ε_n is independent zero-mean Gaussian noise with variance σ^2.

(a) Write down the log likelihood $\log p(y_1, \ldots, y_N | x_1, \ldots, x_N, a, \sigma^2)$ in terms of $y_1, \ldots, y_N, x_1, \ldots, x_N, a, \sigma^2$. [40%]

(b) Assume the following data set of $N = 4$ pairs of points

$$\mathscr{D} = \{(0,1), (1,2), (2,0), (3,4)\}$$

Solve for the maximum likelihood estimates of a and σ^2. [40%]

(c) Assume the same data set, but instead a regression model that predicts x given y:

$$x_n = by_n + \varepsilon_n$$

Is the maximum likehood estimate of b equal to $\frac{1}{a}$? Explain why or why not, giving a derivation if necessary. [40%]

Solution:

(a)

$$\begin{aligned} \log p(y_1, \ldots y_N | x_1, \ldots, x_N, a, \sigma^2) &= \sum_{n=1}^{N} \log p(y_n | x_n, a, \sigma^2) \\ &= -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - ax_n)^2 \end{aligned}$$

(b) Solving for a reduces to minimising

$$(2-a)^2 + (0-2a)^2 + (4-3a)^2$$

Taking derivatives

$$-2(2-a) - 4(0-2a) - 6(4-3a) = 0$$

$$-4+2a+8a-24+18a=0$$

therefore $a=1$. Computing the average squared residuals for σ^2.

$$\sigma^2=\frac{1}{4}[1+1+4+1]=\frac{7}{4}$$

(c) No the ML estimate of b is not $1/a$ since errors are being measured in x now. In fact, minimising $(0-b)^2+(1-2b)^2+(2-0b)^2+(3-4b)^2$ we get $42b=28$, so $b=2/3$.

END OF PAPER