- [ ] Boosting is most effective when the base model has a low bias.
- [ ] Bagging is most effective when the base model has a high bias.
- [ ] Bagging is most effective when the base model has a low bias.

12 Getting Ill During a Pandemic

Your boss is demanding you to come to the office during a pandemic caused by a dangerous virus, despite everyone knowing the chances of contracting the disease during an office visit. It turns out that you have a chance of 20% to contract the virus during a regular workday and 10% during the weekend. Your boss tries to be generous and decides to let you work during the weekend every now and then. He decides on a weekly basis what days you should work, but for now only says, "you'll be working 70% of your days during the weekend".

2.0p **a** You have been working so much lately that you don't even know anymore what day it is, the only thing you know is that your alarm went of and you have to go to the office. Without knowing what day it is, what is your chance of contracting the virus?

Fill in the answer as a percentage, rounded to the nearest integer (so fill in an integer between 0 and 100).

Answer

2.0p **b** Bad luck strikes, you get ill, but luckily you recover well under the doctor's treatment. The doctor would like to know when you have contracted the disease. You have no idea but you could give him the odds that it happened during the weekend. What is the chance that you contracted the virus at work during the weekend?

Fill in the answer as a percentage, rounded to the nearest integer (so fill in an integer between 0 and 100).

Answer

13 K-Means Iterations

Consider the unlabeled dataset $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ of 2D points $x_n \in \mathbb{R}^2$. We want to learn something about the structure of the data and decide to apply the K-means algorithm to split the data in three separate groups ("1", "2", and "3").

Before we do so, let's recap what we know about the K-means algorithm.

1.0p **a** Which of the following statements is true?

- ○ K-means is a semi-supervised learning algorithm
- ○ K-means is an unsupervised learning algorithm
- ○ K-means is a supervised learning algorithm

1.0p **b** Which of the following statements is true?

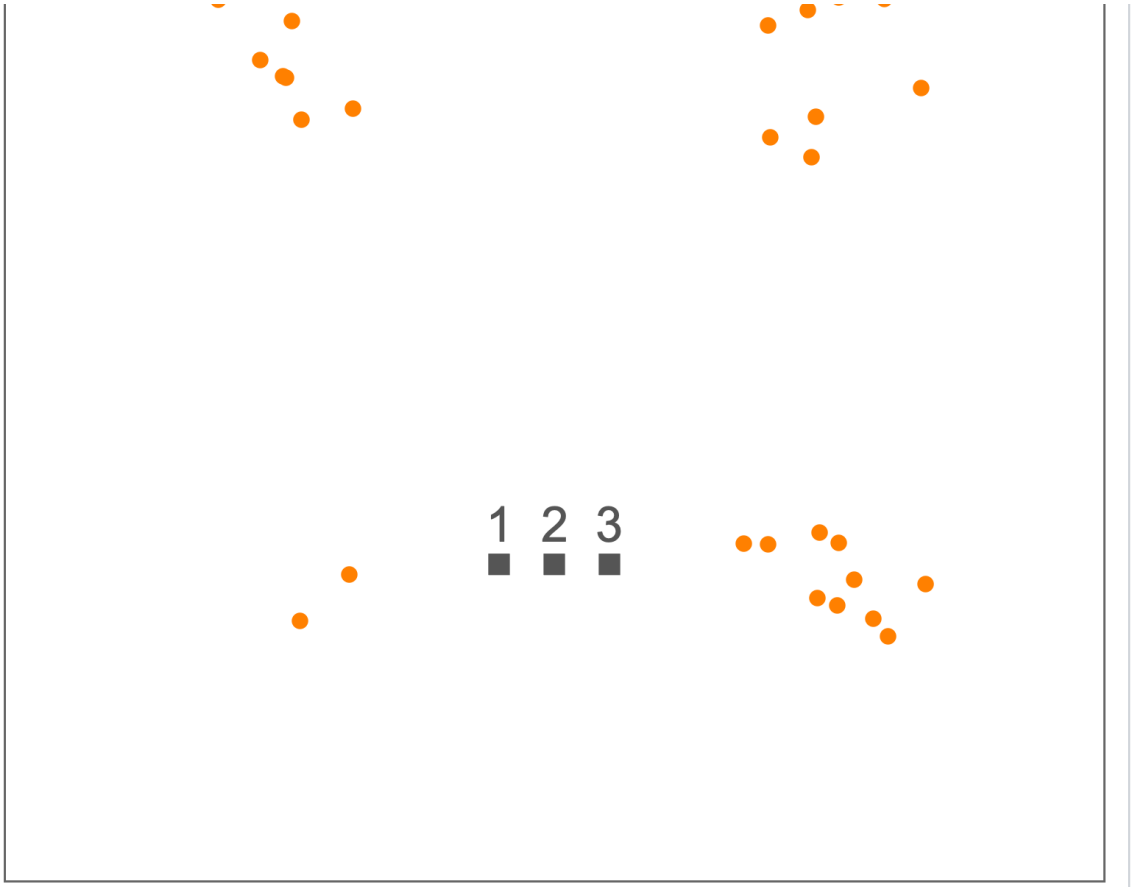
- ☐ The latent variables in K-means are the cluster identities of each data point.
- ☐ The latent variables in K-means are the centroids of the clusters to which a data point can belong.
- ☐ In K-means the data is modeled by a continuous latent variable model
- ☐ In K-means the data is modeled by a discrete latent variable model

3.0p **c** In the figure below, indicate where you expect the cluster centers ("1", "2" and "3") to be after 1000 K-means update iterations, provided that the cluster centers are initialized at the points indicated by the gray rectangle markers.

Select a marker by clicking it

1 2 3

Click in the correct spot to place the marker



2.0p **d** The K-means algorithm can be seen as a version of the E-M algorithm. Explain what happens in the E step and what happens in the M step.

6.0p **e** In EM algorithms the E stands for "Expectation" and the M for "Maximization" and the terminology is typically used in the probabilistic setting when optimizing Mixture Models. Explain why the Expectation/Maximization terminology is also sensible in the K-means clustering algorithm. Do so by indicating the correspondences between steps in the probabilistic and K-means setting.

(It is not necessary to explain the steps in terms of mathematical formulas)

14 Kernel Ridge Regression

We are given the data set $\mathcal{D} = \{\mathbf{x}_i, t_i\}_{i=1}^N$, where each $\mathbf{x}_i \in \mathbb{R}^d$ and $t_i \in \mathbb{R}$. We are also given a collection of feature functions $\{\phi_a(\cdot)\}_{a=1}^K$ that we can use to generate a new feature vector $\boldsymbol{\phi}(\mathbf{x}_i) = (\phi_1(\mathbf{x}_i), \ldots, \phi_K(\mathbf{x}_i))^T \in \mathbb{R}^K$. Now consider the following optimization problem:

$$\min_{\mathbf{w}} \ \sum_{i=1}^N (\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - t_i)^2,$$
$$\text{subject to } \ \|\mathbf{w}\|^2 \leq C.$$

[Latex support]
Some of the math expressions used in this exercise are generated in latex code as follows:

- $\min_{\mathbf{w}} \ \sum_{i=1}^N (\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - t_i)^2,$

\underset{\mathbf{w}}{\operatorname{min}} \;\; \sum_{i=1}^N (\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) - t_i)^2,

- $\|\mathbf{w}\|^2 \leq C.$

\lVert \mathbf{w} \rVert^2 \leq C.

- β

\beta

- Remember to place latex code in one line between dollar signs via `$$code in one line$$`

/[Latex support]

2.0p a Provide an expression for the Lagrangian L. Use β as a symbol to denote the Lagrange multiplier.

(Hint: check that the sign of the Lagrange multiplier is correct)

1.0p **b** Is minimizing the Lagrangian L with respect to $\mathbf{w}$ a convex optimization problem?

- ○ No.
- ○ Yes.

1.0p **c** Explain why it is or isn't a convex optimization problem.

[Latex support]

- $\boldsymbol{\Phi}$

\boldsymbol{\Phi}

/[Latex support]

1.0p **d** Let us define the design matrix $\boldsymbol{\Phi} = \begin{pmatrix} - & \boldsymbol{\phi}(\mathbf{x}_1)^T & - \\ & \vdots & \\ - & \boldsymbol{\phi}(\mathbf{x}_N)^T & - \end{pmatrix}$ which is a $[N \times K]$ matrix. And let us store all target values in the vector $\mathbf{t} = (t_1, \ldots, t_N)^T$. Write down the Lagrangian in terms of the design matrix $\boldsymbol{\Phi}$ and target vector $\mathbf{t}$.

3.0p **e** Derive a closed form expression for the minimizer (with respect to $\mathbf{w}$) of the primal Lagrangian L for a fixed β. Call this minimizer $\mathbf{w}^*(\beta)$.

4.0p **f** Write down all KKT equations (including the optimality condition).

2.0p **g** Assume you are given the minimizer $\mathbf{w}^*(\beta)$ of the Lagrangian L with respect to $\mathbf{w}$ for fixed β. Write down the dual optimizaton problem in terms of $\mathbf{w}^*(\beta)$.

1.0p **h** Is the dual problem convex?

- ◯ no
- ◯ yes

2.0p **i** Why can the provided constrained minimization problem be thought of as a ridge regression problem?

1.0p **j** Let us apply the kernel trick by taking on the dual viewpoint. Using a matrix inversion lemma the solution for $\mathbf{w}^*(\beta)$ can be written as $\mathbf{w}^*(\beta) = \mathbf{\Phi}^T\mathbf{b}$, with $\mathbf{b}$ a dual variable defined by $\mathbf{b} = (\mathbf{K} + \beta\mathbf{I}_N)^{-1}\mathbf{t}$.

Here $\mathbf{K}$ would then be kernel in matrix form. How is it defined? Given an expression for $\mathbf{K}$ in terms of $\mathbf{\Phi}$.

2.0p **k** Assume we have a test case $\mathbf{x}^*$ and you are given a kernel $K(\cdot,\cdot)$ that corresponds to a particular choice of $\boldsymbol{\phi}$. Provide an expression for the predicted value of t using the above ridge regression model. The expression may only involve kernel evaluations (instead of feature evaluations) since you don't know how to compute the features.

2.0p l Assume the kernel does not have any free parameters that you can tune. Now consider a situation where you suspect overfitting. What would you do to reduce this overfitting?

2.0p m Now consider the case of 2D data points, and you choose to work with a kernel that is defined by $k(\mathbf{y}, \mathbf{z}) = (\mathbf{y}^T \mathbf{A} \mathbf{z})^2$, with $\mathbf{A} = \begin{pmatrix} 3 & 0 \\ 0 & 7 \end{pmatrix}$, with $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^2$. What would the corresponding feature transform $\phi(\mathbf{x})$ be?

- ◯ $\phi(x) = (9x_1^2, 37x_1x_2, 49x_2^2)^T$.
- ◯ $\phi(x) = (3x_1^2, \sqrt{3}\sqrt{7}x_1x_2, 7x_2^2)^T$.
- ◯ $\phi(x) = (3x_1^2, 37x_1x_2, 7x_2^2)^T$.
- ◯ $\phi(x) = (7x_1^2, \sqrt{3}\sqrt{7}x_1x_2, 3x_2^2)^T$.

15 Image Content Classification

Consider a database of N images that contain scenes with objects (items, people,

animals, ...) in them. For each image and for each object class we know whether or not it is present and we have annotated the images with a binary vector $\mathbf{t}_n \in \{0,1\}^K$ in which the k^{th} element is 1 if that object corresponding to that index is present and 0 otherwise. An example vector would be $\mathbf{t}_n = (0,1,0,0,1,0,\dots)^T \in \{0,1\}^K$.

We have access to a table of the K expected classes and their class indices. The first 5 classes are listed in the table below.

Class id	Class name
k=1	Adult
k=2	Child
k=3	Cat
k=4	Dog
k=5	Car
...	...

For each image we have access to feature vectors that are precomputed via deep neaural networks. The feature vector for the n^{th} image is denoted with $\mathbf{x}_n \in \mathbb{R}^M$.

All features vectors $\mathbf{x}_n$ are stored in a $N \times M$ matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$. All class indicator vectors $\mathbf{t}_n$ are stored in the $N \times K$ matrix $\mathbf{T} \in \{0,1\}^{N \times K}$.

In this exercise we are interested in retrieving the image content (which object classes are present) for new input images via a probabilistic approach. We model the posterior class probabilities via

(1) $$p(C_k|\mathbf{x}, \mathbf{w}_1, \dots, \mathbf{w}_K) = y_k(\mathbf{x}) = \frac{1}{1+e^{-a_k(\mathbf{x})}}$$

with $a_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x}$, and for each k the weights $\mathbf{w}_k = (w_{k1}, \dots, w_{kM})^T \in \mathbb{R}^M$ are considered to be model parameters.

2.0p a We refer to $p(C_k|\mathbf{x}, \mathbf{w}_1, \dots, \mathbf{w}_K)$ as the *posterior class* probability (as opposed to prior, likelihood, class conditional, naive, ...). Why do we call the above probability the *posterior class* probability?

2.0p **b** Suppose you have already optimized the model weights $\mathbf{w}_1, \ldots, \mathbf{w}_K$ and inspect the posterior class probabilities for a new input image $\mathbf{x}$. You find that $y_3 \approx 0.5$ and $y_4 \approx 0.5$ and all other posterior class probabilities are close to zero, i.e. $\forall_{k \neq 3, k \neq 4} : y_k \approx 0$. What does this tell you about the content of the image?

For the weights we assume a prior distribution which is given by a generalized Gaussian as follows

$$p(\mathbf{w}_1, \ldots, \mathbf{w}_K | \gamma, q) = \prod_{k=1}^{K} \left(\frac{q}{2\,\gamma\,\Gamma(\frac{1}{q})} \right)^M e^{-\|\mathbf{w}_k\|_q^q / \gamma^q} \tag{2}$$

where $\gamma > 0$ is a scale parameter, $q > 0$ determines the shape of the distribution, $\Gamma(\frac{1}{q})$ is some normalization constant, and the q-norm of a vector $\mathbf{w}_k$ (to the power q) is defined as

$$\|\mathbf{w}_k\|_q^q = \sum_{m=1}^{M} |w_{km}|^q.$$

[Latex support]
Some of the math expressions used in this exercise are generated in latex code as follows:

- $p(C_k | \mathbf{x}, \mathbf{w}_1, \ldots, \mathbf{w}_K) = y_k(\mathbf{x}) = \frac{1}{1+e^{-a_k(\mathbf{x})}}$

p(C_k| \mathbf{x}, \mathbf{w}_1, \dots, \mathbf{w}_K) = y_k(\mathbf{x}) = \frac{1}{1 + e^{-a_k(\mathbf{x})}}

- $p(\mathbf{w}_1, \dots, \mathbf{w}_K|\gamma, q) = \prod_{k=1}^{K} \left(\frac{q}{2\,\gamma\,\Gamma(\frac{1}{q})} \right)^M e^{-\|\mathbf{w}_k\|_q^q / \gamma^q}$

p(\mathbf{w}_1,\dots,\mathbf{w}_K | \gamma, q) = \prod_{k=1}^K \left(\frac{q}{2 \, \gamma\, \Gamma(\frac{1}{q})} \right)^M e^{-\lVert \mathbf{w}_k \rVert_q^q / \gamma^q}

- $\|\mathbf{w}_k\|_q^q = \sum_{m=1}^{M} |w_{km}|^q$

\lVert \mathbf{w}_k \rVert_q^q = \sum_{m=1}^{M} | w_{km}|^q

- Remember to place latex code in one line between dollar signs.

[/Latex support]

4.0p **c** Consider two different weight vectors $\mathbf{w}_k$ and $\mathbf{w}_l$ ($k \neq l$).

1. Are they correlated according to the prior in Eq. (2)?
2. Are two different elements of the same weight vector $\mathbf{w}_k$, such as w_{k1} and w_{k2}, correlated?

For both cases, explain your answer.

[Latex support]

Some of the math expressions used in this exercise are generated in latex code as follows:

- $p(\mathbf{w}_1, \dots, \mathbf{w}_K|\mathbf{X}, \mathbf{T}, \gamma, q)$

p(\mathbf{w}_1, \dots, \mathbf{w}_K | \mathbf{X}, \mathbf{T}, \gamma, q)

- $p(\mathbf{T}|\mathbf{X}, \mathbf{w}_1, \dots, \mathbf{w}_K)$

p(\mathbf{T} | \mathbf{X}, \mathbf{w}_1, \dots, \mathbf{w}_K)

- $p(\mathbf{w}_1, \ldots, \mathbf{w}_K | \gamma, q)$

p(\mathbf{w}_1, \dots, \mathbf{w}_K | \gamma, q)
[/Latex support]

2.0p **d** Write down the expression for the posterior $p(\mathbf{w}_1, \ldots, \mathbf{w}_K | \mathbf{X}, \mathbf{T}, \gamma, q)$ in terms of the data likelihood $p(\mathbf{T} | \mathbf{X}, \mathbf{w}_1, \ldots, \mathbf{w}_K)$ and the prior $p(\mathbf{w}_1, \ldots, \mathbf{w}_K | \gamma, q)$. You do not need to explicitly write out the actual distributions.

3.5p **e** Give the expressions for

1. the log-likelihood $\ln p(t_n | \mathbf{x}_n, \mathbf{w}_1, \ldots, \mathbf{w}_K)$ of a single training example ($\mathbf{x}_n$ with corresponding $\mathbf{t}_n$) given the probabilistic model of equation (1) **in terms of** y_k,
2. the log-likelihood for the full dataset $\ln p(\mathbf{T} | \mathbf{X}, \mathbf{w}_1, \ldots, \mathbf{w}_K)$ **in terms of** y_k and
3. the log of the prior $\ln p(\mathbf{w}_1, \ldots, \mathbf{w}_K | \gamma, q)$.

[Latex support]

- $L(\mathbf{w}_1, \ldots, \mathbf{w}_K, \mu) = \sum_{n=1}^{N} \sum_{k=1}^{K} \left[-t_{nk} \ln y_k(\mathbf{x}_n) - (1 - t_{nk}) \ln(1 - y_k(\mathbf{x}_n)) + \mu \|\mathbf{w}_k\|_q^q \right]$

L(\mathbf{w}_1, \dots, \mathbf{w}_K, \mu) = \sum_{n=1}^N \sum_{k=1}^K \left[\, -t_{nk} \ln y_k(\mathbf{x}_n) - (1 - t_{nk})\ln(1 - y_k(\mathbf{x}_n)) + \mu\lVert \mathbf{w}_k \rVert_q^q \, \right]
[/Latex support]

[/Latex support]

3.0p **f** Show that the optimization problem corresponding to obtaining a Maximum A Posteriori (MAP) estimate for $\mathbf{w}_1, \ldots, \mathbf{w}_K$ is equivalent to minimizing the corresponding loss function

$$L(\mathbf{w}_1, \ldots, \mathbf{w}_K, \mu) = \sum_{n=1}^{N} \sum_{k=1}^{K} \left[-t_{nk} \ln y_k(\mathbf{x}_n) - (1 - t_{nk}) \ln(1 - y_k(\mathbf{x}_n)) + \mu \|\mathbf{w}_k\|_q^q \right]$$

with respect to $\mathbf{w}_1, \ldots, \mathbf{w}_K$, and with μ a regularization penalty parameter. Also explain how μ is related to γ and q?

3.0p **g** After reinspecting the dataset we discover that, quite unexpectedly, and perhaps disturbingly, many of the images feature clowns. We decide to go over all images again and add the "clown" class to all the target vectors $\mathbf{t}_n$, such that we can retrain our model and make it capable of detecting the presence of clowns in images. The new target vectors $\mathbf{t}_n \in \{0, 1\}^{K+1}$ are now thus of dimension $K + 1$. Do we need to retrain the entire model? Explain your answer.

2.0p **h** Let us finally take a look again at the prior on the weights $\mathbf{w}_k$ as given in equation (2), and see what it looks like for one particular weight vector $\mathbf{w}$ in the 1D and 2D case, for which $\mathbf{w} \in \mathbb{R}$ or $\mathbf{w} \in \mathbb{R}^2$ respectively. In 1D, the generalized Gaussian is given by

$$p(w|\gamma, q) = \frac{q}{2\gamma\Gamma(\frac{1}{q})} e^{-|w|^q/\gamma^q}.$$

In the figure below it is plotted for the 1D case (left) and the 2D case (right). The left plot directly shows the prior probabilities for some value w given the model parameters $\gamma = 1$ and several values for q. In the right figure you see the contour plot for the 2D distributions, showing the level set $|w_1|^q + |w_2|^q = C$ for some arbitrary value C.

Imagine that you train several models with MAP estimates for $\mathbf{w}_1, \ldots, \mathbf{w}_K$ for different values of $q > 0$ (and $\gamma = 1$).

1. Which trained model will have sparse weight vectors, the ones with $q > 1$ or those with $q \leq 1$? Explain your answer.
2. Given the correct choice for q, how should the γ parameter be changed in order to increase the sparsity of the solutions for the weights $\mathbf{w}_k$?

(Remember, the more elements in a vector take on the value 0, the sparser we consider it to be).