A. Kernel PCA

In this problem we will analyze a hypothesis set based on KPCA projection. Let $K(x, y)$ be a kernel function, $\Phi_K(x)$ be its corresponding feature map and $S = \{x_1, \ldots, x_m\}$ be a sample of m points. When Π is the rank-r KPCA projection, we define the (regularized) hypothesis set of linear separators in the RKHS $\mathbb{H}$ of kernel K as

$$H = \left\{ x \to \langle w, \Pi\Phi_K(x)\rangle_{\mathbb{H}} : \|w\|_{\mathbb{H}} \leq 1 \right\}. \tag{1}$$

This hypothesis set essentially means that the input data is projected onto a smaller dimensional subspace of the RKHS before fitting a separation hyperplane. This problem will show that we can use the eigenvectors and eigenvalues of the sample kernel matrix to give a closed form expression for the functions $h \in H$ without a need for explicit representation of the RKHS itself.

Let $\mathbf{K}$ be the sample kernel matrix for kernel K evaluated on m points of sample S, that is $\mathbf{K}_{i,j} = K(x_i, x_j)$. Let $\lambda_1, \ldots, \lambda_r$ be the top r (nonzero) eigenvalues of $\mathbf{K}$ with the corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$. Denote the j-th element of vector $\mathbf{v}_i$ as $[\mathbf{v}_i]_j$. Follow the subproblems below to derive the explicit representation of $h \in H$.

1. Assume that the feature maps $\Phi_K(x)$ are centered on sample S and recall that the sample covariance operator is $\Sigma = \sum_{i=1}^{m} \frac{1}{m} \Phi_K(x_i)\Phi_K(x_i)^\top$. Prove that $h(x) = \sum_{i=1}^{r} \alpha_i \langle \mathbf{u}_i, \Phi_K(x)\rangle_{\mathbb{H}}$ for some $\alpha_i \in \mathbb{R}$, where $\mathbf{u}_1, \cdots, \mathbf{u}_r$ are the eigenvectors of Σ corresponding to its top r eigenvalues.

   Since w is a function in the r-dimensional subspace of the RKHS, it can be written in the following form

   $$w = \sum_{j=1}^{r} a_j \Phi_K(x_j) \ , \ a_j \in \mathbb{R}$$

   And by definition of the KPCA projection, $\Pi\Phi_K(x)$ takes the following form

   $$\begin{aligned} \Pi\Phi_K(x) &= \mathbf{U}^\top \Phi_K(x) \\ &= \sum_{i=1}^{r} \mathbf{u}_i \Phi_K(x_i)(x) \\ &= \sum_{i=1}^{r} \mathbf{u}_i K(x_i, x) \end{aligned}$$

   Which also coincides with the expression of a function in the r-dimensional subspace of the RKHS. Therefore, we can write the inner product as follows:

   $$\begin{aligned} \langle w, \Pi\Phi_K(x)\rangle_{\mathbb{H}} &= \sum_{i=1}^{r}\sum_{j=1}^{r} K(x_i, x) a_j K(\mathbf{u}_i, \Phi_K(x_j)) \\ &= \sum_{i=1}^{r} \alpha_i \langle \mathbf{u}_i, \Phi_K(x)\rangle \quad (Q.E.D.) \end{aligned}$$

2. Prove that $\mathbf{u}_i = \mathbf{X}\frac{\mathbf{v}_i}{\sqrt{\lambda_i}}$, where $\mathbf{X} = [\Phi_K(x_1), \ldots, \Phi_K(x_m)]$

   Based on the definition of $\mathbf{X}$, we can write $\mathbf{K}$ and $\mathbf{\Sigma}$ as follows:

   $$\mathbf{K} = [K(x_i, x_j)]_{ij} = [\langle \Phi_K(x_i), \Phi_K(x_j)\rangle]_{ij} = \mathbf{X}^\top \mathbf{X}$$

   $$\mathbf{\Sigma} = \sum_{i=1}^{m} \frac{1}{m} \Phi_K(x_i)\Phi_K(x_i)^\top = \frac{1}{m}\sum_{i=1}^{m} \mathbf{X}\mathbf{X}^\top$$

   If we consider the singular value decomposition (SVD) of $\mathbf{X}$:

   $$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\top$$

and use the basic property in linear algebra for the transpose of a product of matrices:

$$\left(\prod_{i=1}^{N} \mathbf{A}_i\right)^\top = \prod_{i=0}^{N-1} \mathbf{A}_{N-i}^\top$$

we can simplify the expressions for $\mathbf{K}$ and $\mathbf{\Sigma}$:

$$\mathbf{K} = (\mathbf{USV}^\top)^\top \mathbf{USV}^\top = \mathbf{VS}^2\mathbf{V}^\top$$
$$\mathbf{\Sigma} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{USV}^\top(\mathbf{USV}^\top)^\top = \frac{1}{m}\mathbf{US}^2\mathbf{U}^\top$$

Which is a direct result from the fact that $\mathbf{U}$ and $\mathbf{V}$ are unitary and $\mathbf{S}$ is diagonal according to the definition of SVD decomposition.
If we look at the expression for $\mathbf{\Sigma}$ we notice that it takes the form of an SVD, with matrix of eigenvectors $\mathbf{U}$ and matrix of eigenvalues $\mathbf{\Lambda} = \mathbf{S}^2$. Similarly for $\mathbf{K}$, where the matrix of eigenvectors is $\mathbf{V}$ and the matrix of eigenvalues is the same $\mathbf{\Lambda}$.
Now we return to the SVD expression for $\mathbf{X}$ and do the following manipulations to derive an expression for $\mathbf{U}$:

$$\mathbf{USV}^\top = \mathbf{X}$$
$$\mathbf{USV}^\top\mathbf{VS}^{-1} = \mathbf{XVS}^{-1}$$
$$\mathbf{U} = \mathbf{XV}\mathbf{\Lambda}^{-\frac{1}{2}}$$

We can rewrite this last expression in its vector form, and since we now that $\mathbf{V}$ and $\mathbf{\Lambda}$ contain the eigenvectors and eigenvalues of $\mathbf{K}$, we get the desired result:

$$\mathbf{u}_i = \mathbf{X}\frac{\mathbf{v}_i}{\sqrt{\lambda_i}} \quad (Q.E.D.)$$

3. Using the result above, prove that any function $h \in H$ can be represented as

$$h(x) = \sum_{i=1}^{r}\sum_{j=1}^{m} \frac{\alpha_i}{\sqrt{\lambda_i}} K(x_j, x)[v_i]_j,$$

for some $\alpha_i \in \mathbb{R}$.
With the definition of $\mathbf{X} = [\Phi_K(x_1), \ldots, \Phi_K(x_m)]$, we can rewrite the previous result as follows:

$$\mathbf{u}_i = \frac{1}{\sqrt{\lambda_i}}\sum_{j=1}^{m} \Phi_K(x_j)[\mathbf{v}_i]_j$$

We plug in this result in the expression for $h(x)$ from the first exercise and simplify:

$$\begin{aligned}
h(x) &= \sum_{i=1}^{r} \alpha_i \langle \mathbf{u}_i, \Phi_K(x)\rangle_{\mathbb{H}} \\
&= \sum_{i=1}^{r} \alpha_i \left\langle \frac{1}{\sqrt{\lambda_i}}\sum_{j=1}^{m} \Phi_K(x_j)[\mathbf{v}_i]_j, \Phi_K(x) \right\rangle_{\mathbb{H}} \\
&= \sum_{i=1}^{r} \frac{\alpha_i}{\sqrt{\lambda_i}} \left\langle \sum_{j=1}^{m} \Phi_K(x_j)[\mathbf{v}_i]_j, \Phi_K(x) \right\rangle_{\mathbb{H}} \\
&= \sum_{i=1}^{r} \frac{\alpha_i}{\sqrt{\lambda_i}} \sum_{j=1}^{m} \langle \Phi_K(x_j)[\mathbf{v}_i]_j, \Phi_K(x)\rangle_{\mathbb{H}} \\
&= \sum_{i=1}^{r}\sum_{j=1}^{m} \frac{\alpha_i}{\sqrt{\lambda_i}} [\mathbf{v}_i]_j \langle \Phi_K(x_j), \Phi_K(x)\rangle_{\mathbb{H}} \\
&= \sum_{i=1}^{r}\sum_{j=1}^{m} \frac{\alpha_i}{\sqrt{\lambda_i}} K(x_j, x)[\mathbf{v}_i]_j \quad (Q.E.D.)
\end{aligned}$$

All the manipulations of the inner product are straightforward due to the linearity in the first argument, and the last substitution is simply the definition of the kernel function K.

4. Bonus question: derive the Rademacher complexity bound on the hypothesis set H defined in this problem.
By definition of the Rademacher complexity we write:

$$\begin{aligned}\widehat{\mathfrak{R}}_S(H) &= \underset{\sigma}{E}\left[\sup_{h\in H}\left(\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i)\right)\right] \\ &= \frac{1}{m}\underset{\sigma}{E}\left[\sup_{\|w\|\leq 1}\left(\sum_{i=1}^{m}\sigma_i \left\langle w, \Pi\Phi_K(x_i)\right\rangle\right)\right]\end{aligned}$$

Since we are considering real variables, the inner product is linear in its second argument. Also, we introduce an absolute value ($\sup A \leq \sup |A|$ is straightforward) so we can apply the Cauchy-Schwartz inequality:

$$\begin{aligned}\widehat{\mathfrak{R}}_S(H) &\leq \frac{1}{m}\underset{\sigma}{E}\left[\sup_{\|w\|\leq 1}\left|\left\langle w, \sum_{i=1}^{m}\sigma_i \Pi\Phi_K(x_i)\right\rangle\right|\right] \\ &\leq \frac{1}{m}\underset{\sigma}{E}\left[\sup_{\|w\|\leq 1}||w||\cdot\left|\left|\sum_{i=1}^{m}\sigma_i \Pi\Phi_K(x_i)\right|\right|\right]\end{aligned}$$

Clearly the supremum occurs when $||w|| = 1$ and therefore we get

$$\widehat{\mathfrak{R}}_S(H) \leq \frac{1}{m}\underset{\sigma}{E}\left[\left|\left|\sum_{i=1}^{m}\sigma_i \Pi\Phi_K(x_i)\right|\right|\right]$$

Next we introduce a square root function (concave) to apply Jensen's inequality, and expand the squared norm of the sum:

$$\begin{aligned}\widehat{\mathfrak{R}}_S(H) &\leq \frac{1}{m}\underset{\sigma}{E}\left[\sqrt{\left|\left|\sum_{i=1}^{m}\sigma_i \Pi\Phi_K(x_i)\right|\right|^2}\right] \\ &\leq \frac{1}{m}\left[\underset{\sigma}{E}\left[\left|\left|\sum_{i=1}^{m}\sigma_i \Pi\Phi_K(x_i)\right|\right|^2\right]\right]^{\frac{1}{2}} \\ &= \frac{1}{m}\left[\underset{\sigma}{E}\left[\sum_{i=1}^{m}||\sigma_i \Pi\Phi_K(x_i)||^2 + \sum_{i\neq j}\left\langle\sigma_i \Pi\Phi_K(x_i), \sigma_j \Pi\Phi_K(x_j)\right\rangle\right]\right]^{\frac{1}{2}} \\ &= \frac{1}{m}\left[\underset{\sigma}{E}\left[\sum_{i=1}^{m}\sigma_i^2 ||\Pi\Phi_K(x_i)||^2\right] + \underset{\sigma}{E}\left[\sum_{i\neq j}\sigma_i\sigma_j \left\langle\Pi\Phi_K(x_i), \Pi\Phi_K(x_j)\right\rangle\right]\right]^{\frac{1}{2}} \\ &= \frac{1}{m}\left[\sum_{i=1}^{m}||\Pi\Phi_K(x_i)||^2\right]^{\frac{1}{2}}\end{aligned}$$

For the last step, we notice in the first term that $\sigma_i^2 = 1$ (since all the σs are Rademacher variables). Also the second term will reduce to zero, since $E[\sigma_i\sigma_j] = 0$ for $i \neq j$ (again, this follows from the fact that the σs are Rademacher variables).
At this point we are left with the expression $\Pi\Phi_K(x_i)$, which by definition is the projection of the input data point x_i over a subspace of the RKHS of dimension r . From equation (12.5) in

the textbook, we know that this projection will take the following form

$$\begin{aligned}\Pi\Phi_K(x_i) &= \Phi_K(x_i)^\top \mathbf{U}_r \\ &= \sqrt{\lambda_i}\mathbf{v}_{ir}\end{aligned}$$

Where $\mathbf{U}_r$ stands for the first r columns of matrix $\mathbf{U}$ and $\mathbf{v}_{ir}$ stands for the first r elements of vector $\mathbf{v}_i$. Since the vectors $\mathbf{v}_i$ by definition have norm 1, it must follow that $||\mathbf{v}_{ir}|| \leq 1$ and we write:

$$\begin{aligned}\mathfrak{R}_S(H) &\leq \frac{1}{m}\left[\sum_{i=1}^{m}\left|\left|\sqrt{\lambda_i}\mathbf{v}_{ir}\right|\right|^2\right]^{\frac{1}{2}} \\ &\leq \frac{1}{m}\left[\sum_{i=1}^{m}\lambda_i\right]^{\frac{1}{2}}\end{aligned}$$

Finally, we can express the bound in two different ways:

(a) It is a known result in linear algebra that the sum of the eigenvalues of a matrix is equal to its trace. In this particular case, the trace of $\mathbf{K}$ has elements of the form $K(x_i, x_i)$. Let R be the largest of these elements and we write

$$\begin{aligned}\mathfrak{R}_S(H) &\leq \frac{1}{m}\left[\sum_{i=1}^{m}K(x_i, x_i)\right]^{\frac{1}{2}} \\ &\leq \frac{1}{m}\sqrt{mR} \\ &= \sqrt{\frac{R}{m}}\end{aligned}$$

(b) Since the eigenvalues λ_i are ranked in decreasing order of magnitude, we know that $\lambda_i \leq \lambda_1$ for all i and we write

$$\begin{aligned}\mathfrak{R}_S(H) &\leq \frac{1}{m}\left[\sum_{i=1}^{m}\lambda_i\right]^{\frac{1}{2}} \\ &\leq \frac{1}{m}\sqrt{m\lambda_1} \\ &= \sqrt{\frac{\lambda_1}{m}}\end{aligned}$$

B. Multi-class boosting

Lecture 10 introduces the AdaBoost.MH algorithm, which is AdaBoost for multi-class classification. (Consult with Lecture 10's slides if you are unfamiliar with multi-class learning setting.) AdaBoost.MH is defined by objective function $F(\alpha)$:

$$F(\alpha) = \sum_{l=1}^{k} \sum_{i=1}^{m} e^{-y_i[l] \sum_{t=1}^{n} \alpha_t h_t(x_i,l)},$$

where $y_i \in \mathcal{Y} = \{-1, +1\}^k$, and $y_i[l]$ denotes the l-th coordinate of y_i for any $i \in [m]$ and $l \in [k]$. The base classifiers come from $H = \{h : \mathcal{X} \times [k] \to \{-1, +1\}\}$. Consider an alternative objective function for the same problem:

$$G(\alpha) = \sum_{i=1}^{m} e^{-\frac{1}{k} \sum_{l=1}^{k} y_i[l] \sum_{t=1}^{n} \alpha_t h_t(x_i,l)}.$$

1. Compare $G(\alpha)$ with $F(\alpha)$. Show that $F(\alpha) \geq kG(\alpha)$.
   First we define the function $g_n(x_i, l) = \sum_{t=1}^{n} \alpha_t h_t(x_i, l)$ and rewrite $F(\alpha)$ and $G(\alpha)$ as follows

   $$F(\alpha) = \sum_{l=1}^{k} \sum_{i=1}^{m} e^{-y_i[l] g_n(x_i,l)}$$
   $$G(\alpha) = \sum_{i=1}^{m} e^{-\frac{1}{k} \sum_{l=1}^{k} y_i[l] g_n(x_i,l)}$$

   If we interchange the order of summations in F and we can prove that the inequality holds for any value of i, then it must also hold for the entire summation. So the problem reduces to proving that

   $$\sum_{l=1}^{k} e^{-y_i[l] g_n(x_i,l)} \geq k e^{-\frac{1}{k} \sum_{l=1}^{k} y_i[l] g_n(x_i,l)}$$
   $$\frac{1}{k} \sum_{l=1}^{k} e^{-y_i[l] g_n(x_i,l)} \geq e^{-\frac{1}{k} \sum_{l=1}^{k} y_i[l] g_n(x_i,l)}$$
   $$E\left[e^{-y_i[l] g_n(x_i,l)}\right] \geq e^{E[-y_i[l] g_n(x_i,l)]}$$

   Since the exponential function is convex, the last expression coincides with Jensen's inequality and the proof is complete.

2. Let $g_n(x_i, l) = \sum_{t=1}^{n} \alpha_t h_t(x_i, l)$. Assume that $|g_n(x_i, l)| \leq 1$ for all $x_i \in \mathcal{X}, l \in [k]$. Show that $kG(\alpha)$ is a convex function upper bounding the multi-label multi-class error:

   $$\sum_{i=1}^{m} \sum_{l=1}^{k} 1_{y_i[l] \neq sgn(g_i(x_i,l))} \leq kG(\alpha).$$

   It is clear that $kG(\alpha)$ is a convex function since G is defined as a sum of exponential (convex) functions and $k > 0$. Similarly to the previous exercise, we can get rid of the summation over i and only need to prove the following inequality

   $$\sum_{l=1}^{k} 1_{y_i[l] \neq sgn(g_n(x_i,l))} \leq k e^{-\frac{1}{k} \sum_{l=1}^{k} y_i[l] g_n(x_i,l)}$$

   We will start by proving that

   $$\sum_{l=1}^{k} y_i[l] g_n(x_i, l) \leq k - r$$

   where $r \in [0, k]$ is the number of mistakes made by the classifier.

- By definition the classifier predicts correctly if $y_i[l] = sgn(g_n(x_i, l))$, which means that in the previous sum positive values of $y_i[l]g_n(x_i, l)$ are associated to correct predictions and vice versa.
- Since we know that r is the number of errors the classifier made, the following is an upper bound on the value for the sum:

$$(k-r) \cdot \max_{l:\text{correct}} (y_i[l]g_n(x_i, l)) + r \cdot \max_{l:\text{mistake}} (y_i[l]g_n(x_i, l))$$

- Finally, since we know that $|g_n(x_i, l)| \leq 1$, the maximum value for the sum is given by $(k-r) \cdot 1 + r \cdot 0 = k - r$.

Now we can easily complete the proof:

$$\begin{aligned}
\sum_{l=1}^{k} y_i[l]g_n(x_i, l) &\leq k - r \\
-\frac{1}{k}\sum_{l=1}^{k} y_i[l]g_n(x_i, l) &\geq \frac{r-k}{k} \\
ke^{-\frac{1}{k}\sum_{l=1}^{k} y_i[l]g_n(x_i,l)} &\geq ke^{\frac{r-k}{k}} \\
&\geq k\left(1 + \frac{r-k}{k}\right) \\
&\geq r \\
&= \sum_{l=1}^{k} 1_{y_i[l] \neq sgn(g_n(x_i,l))} \quad (Q.E.D.)
\end{aligned}$$

3. Drive an algorithm defined by the application of coordinate descent to $G(\alpha)$. You should give a full description of your algorithm, including the pseudocode, details for the choice of the step and direction, as well as a generalization bound.

ADABOOST.MH2($S = ((x_1, y_1), \ldots, (x_m, y_m))$)

for $i \leftarrow 1$ **to** m **do**
  $D_1(i) \leftarrow \frac{1}{m}$
for $t \leftarrow 1$ **to** T **do**
  $h_t \leftarrow$ base classifier in H with small error $\epsilon_t = \Pr_{(i,l)\sim D_t}[h_t(x_i, l) \neq y_i[l]]$
  $\alpha_t \leftarrow \frac{k}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
  $Z_t \leftarrow \left[2\sqrt{\epsilon_t(1-\epsilon_t)}\right]^k$
  for $i \leftarrow 1$ **to** m **do**
    $S_t(i) \leftarrow \sum_{l=1}^{k} y_i[l]h_t(x_i, l)$
    $D_{t+1}(i) \leftarrow \frac{D_t(i)}{Z_t} \cdot \exp\left(-\frac{\alpha_t}{k}S_t(i)\right)$
$g \leftarrow \sum_{t=1}^{T} \alpha_t h_t$
return $h = \text{sgn}(g)$

This algorithm is very similar to the original ADABOOST for binary classification. Besides from the introduction of the auxiliary function $S_t(i) = \sum_{l=1}^{k} y_i[l]h_t(x_i, l)$, which we will reference throughout the analysis, the main differences are the way of calculating α_t and the normalization factor Z_t. Applying coordinate descent to the objective function $G(\boldsymbol{\alpha})$ will provide a justification for the selection of these parameters.

The expression that we will be using for $G(\alpha)$ is the following:

$$\begin{aligned} G(\alpha) &= \sum_{i=1}^{m} e^{-\frac{1}{k}\sum_{l=1}^{k} y_i[l] \sum_{t=1}^{T} \alpha_t h_t(x_i,l)} \\ &= \sum_{i=1}^{m} e^{-\frac{1}{k}\sum_{t=1}^{T} \alpha_t \sum_{l=1}^{k} y_i[l] h_t(x_i,l)} \\ &= \sum_{i=1}^{m} e^{-\frac{1}{k}\sum_{t=1}^{T} \alpha_t S_t(i)} \end{aligned}$$

(a) **Choice of the direction**
Performing a similar analysis as the one presented in the textbook, we find the direction $\mathbf{e}_t$ selected by coordinate descent:

$$\begin{aligned} \mathbf{e}_t &= \underset{t}{\operatorname{argmin}} \left. \frac{dG(\boldsymbol{\alpha}_{t-1} + \eta \mathbf{e}_t)}{d\eta} \right|_{\eta=0} \\ G(\boldsymbol{\alpha}_{t-1} + \eta \mathbf{e}_t) &= \sum_{i=1}^{m} \exp\left[-\frac{1}{k} \sum_{s=1}^{t-1} \alpha_s S_t(i) - \frac{\eta}{k} S_t(i) \right] \\ \left. \frac{dG}{d\eta} \right|_{\eta=0} &= -\frac{1}{k} \sum_{i=1}^{m} S_t(i) \exp\left[-\frac{1}{k} \sum_{s=1}^{t-1} \alpha_s S_t(i) \right] \\ &= -\frac{1}{k} \sum_{i=1}^{m} S_t(i) \left(D_t(i) \cdot m \prod_{s=1}^{t-1} Z_s \right) \\ &= \left(-\frac{m}{k} \prod_{s=1}^{t-1} Z_s \right) \sum_{i=1}^{m} S_t(i) D_t(i) \end{aligned}$$

To simplify the last expression, we will rewrite the summation over all values of i splitting it into "bins". In each bin we will group all the data points such that $S_t(i) = k - 2r$, where $r \in [0, k]$ is the number of errors that the base classifier h_t makes over all labels. Considering that for each label the base classifier makes a mistake with small probability ϵ_t we can write

$$\Pr[\#\text{Errors} = r] = \Pr[S_t(i) = k - 2r] = \binom{k}{r} \epsilon_t^r (1 - \epsilon_t)^{k-r}$$

Now we can use this distribution and sum over all values of r as an equivalent to the distribution D_t over all values of i. In the original version of *AdaBoost* for binary classification this part of the analysis corresponds to splitting the sample points into correctly and incorrectly classified.
Returning to the expression for the derivative of G we get

$$\begin{aligned} \left. \frac{dG}{d\eta} \right|_{\eta=0} &= \left(-\frac{m}{k} \prod_{s=1}^{t-1} Z_s \right) \sum_{r=0}^{k} \binom{k}{r} \epsilon_t^r (1 - \epsilon_t)^{k-r} S_t(i) \\ &= \left(-\frac{m}{k} \prod_{s=1}^{t-1} Z_s \right) \sum_{r=0}^{k} \binom{k}{r} \epsilon_t^r (1 - \epsilon_t)^{k-r} (k - 2r) \\ &= \left(-\frac{m}{k} \prod_{s=1}^{t-1} Z_s \right) (k - 2k\epsilon_t) \\ &= \left(m \prod_{s=1}^{t-1} Z_s \right) (2\epsilon_t - 1) \end{aligned}$$

Using *Wolfram Mathematica* we get the closed form for the summation and arrive to the exact same result than for the original *AdaBoost* algorithm. Since $m \prod_{s=1}^{t-1} Z_s$ is fixed and positive, the direction $\mathbf{e}_t$ selected by coordinate descent is the one minimizing ϵ_t, which corresponds to the base learner h_t.

(b) **Choice of the step**
In a similar fashion we can find the optimum value for the step size η by setting the derivative of G equals to 0 and solving:

$$\begin{aligned}
\frac{dG}{d\eta} &= 0 \\
\frac{d}{d\eta}\left[\sum_{i=1}^{m}\exp\left[-\frac{1}{k}\sum_{s=1}^{t-1}\alpha_s S_t(i) - \frac{\eta}{k}S_t(i)\right]\right] &= 0 \\
\sum_{i=1}^{m}\exp\left[-\frac{1}{k}\sum_{s=1}^{t-1}\alpha_s S_t(i)\right]\exp\left[-\frac{\eta}{k}S_t(i)\right]\left(-\frac{S_t(i)}{k}\right) &= 0 \\
\sum_{i=1}^{m}\left(D_t(i)\cdot m\prod_{s=1}^{t-1}Z_s\right)\exp\left[-\frac{\eta}{k}S_t(i)\right]\left(-\frac{S_t(i)}{k}\right) &= 0 \\
\sum_{r=0}^{k}\binom{k}{r}\epsilon_t^r(1-\epsilon_t)^{k-r}e^{-\frac{k-2r}{k}\eta}(k-2r) &= 0 \\
ke^{-\eta}\left[(1-\epsilon_t)-\epsilon_t e^{\frac{2\eta}{k}}\right]\left[(1-\epsilon_t)+\epsilon_t e^{\frac{2\eta}{k}}\right]^{k-1} &= 0 \\
\eta &= \frac{k}{2}\log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)
\end{aligned}$$

Again we use *Wolfram Mathematica* to get the closed form for the summation. In the last step we notice that all the factors are strictly positive except for $\left[(1-\epsilon_t)-\epsilon_t e^{\frac{2\eta}{k}}\right]$. Setting this factor to 0 and solving for η gives us the desired value, which coincides with the selection made in *AdaBoost.MH2* for the parameter α_t.

(c) **Generalization bound**
Finally, we derive an upper bound on the emprical error of *AdaBoost.MH2*. Using the result from exercise 2 we write

$$\begin{aligned}
\widehat{R}(h) &= \frac{1}{m}\sum_{i=1}^{m}\sum_{l=1}^{k}1_{y_i[l]\neq sgn\left(\sum_{t=1}^{T}\alpha_t h_t(x_i,l)\right)} \\
&\leq \frac{k}{m}G(\alpha) \\
&= \frac{k}{m}\sum_{i=1}^{m}\exp\left[-\frac{1}{k}\sum_{t=1}^{T}\alpha_t S_t(i)\right] \\
&= \frac{k}{m}\sum_{i=1}^{m}D_t(i)\cdot m\prod_{t=1}^{T}Z_t \\
&= k\prod_{t=1}^{T}Z_t
\end{aligned}$$

All that's left is to find a closed form for the product of the regularization factors Z_t:

$$\begin{aligned}
Z_t &= \sum_{i=1}^{m} D_t(i) e^{-\frac{1}{k}\alpha_t S_t(i)} \\
&= \sum_{r=0}^{k} \binom{k}{r} \epsilon_t^r (1-\epsilon_t)^{k-r} e^{-\frac{k-2r}{k}\alpha_t} \\
&= \left[(1-\epsilon_t)e^{-\frac{\alpha_t}{k}} + \epsilon_t e^{\frac{\alpha_t}{k}}\right]^k \\
&= \left[(1-\epsilon_t)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}} + \epsilon_t\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}\right]^k \\
&= \left[2\sqrt{\epsilon_t(1-\epsilon_t)}\right]^k
\end{aligned}$$

Plugging this value into the upper bound for $\widehat{R}(h)$ we get

$$\begin{aligned}
\widehat{R}(h) &\leq k \prod_{t=1}^{T} \left[2\sqrt{\epsilon_t(1-\epsilon_t)}\right]^k \\
&= k \left[\prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)}\right]^k \\
&= k \left[\prod_{t=1}^{T} \sqrt{1 - 4\left(\frac{1}{2} - \epsilon_t\right)^2}\right]^k \\
&= k \left[\prod_{t=1}^{T} \exp\left[-2\left(\frac{1}{2} - \epsilon_t\right)^2\right]\right]^k \\
&\leq k \left[\exp\left[-2\sum_{t=1}^{T}\left(\frac{1}{2} - \epsilon_t\right)^2\right]\right]^k \\
&= k \exp\left[-2k\sum_{t=1}^{T}\left(\frac{1}{2} - \epsilon_t\right)^2\right]
\end{aligned}$$

The analysis is basically the same as the one used to prove Theorem 6.1 from the textbook.