10.1 (f)

We first show

10.1 (c)

$k(x,z) = k_1(x,z)\, k_2(x,z)$ is a valid kernel

Suppose the feature map for k_1 and k_2 are:

$$\phi^1(x) = [\phi_1^1(x), \phi_2^1(x), \dots, \phi_{N_1}^1(x)]$$

$$\phi^2(x) = [\phi_1^2(x), \dots\dots, \phi_{N_2}^2(x)]$$

Consider a feature map

$$\phi(x) = [\phi_1^1(x)\phi_1^2(x), \phi_1^1(x)\phi_2^2(x), \dots, \phi_1^1(x)\phi_{N_2}^2(x), \dots, \phi_{N_1}^1(x)\phi_{N_2}^2(x)]$$

$$\Rightarrow \phi(x)^T \phi(z)$$

$$= \phi_1^1(x)\phi_1^2(x) \cdot \phi_1^1(z)\phi_1^2(z) + \dots + \phi_{N_1}^1(x)\phi_{N_2}^2(x)\, \phi_{N_1}^1(z)\phi_{N_2}^2(z)$$

$$= \phi_1^1(x)\phi_1^1(z)\, \phi_1^2(x)\phi_1^2(z) + \dots + \phi_{N_1}^1(x)\phi_{N_1}^1(z)\phi_{N_2}^2(x)\,\phi_{N_2}^2(z)$$

$$= \phi^1(x)^T \phi^1(z) \cdot \phi^2(x)^T \cdot \phi^2(z)$$

$$= k_1(x,z) \cdot k_2(x,z) \qquad ①$$

Then we try to show that

$k(x,z) = k_1(x,z)^q$ is a valid kernel

By ①, we know $k'(x,z) = k_1(x,z)^2$ is a valid kernel

$\Rightarrow$ $k''(x,z) = k'(x,z) \cdot k_1(x,z) = k^3(x,z)$

is a valid kernel

$\Rightarrow$ By induction, we know that

$k(x,z) = k_1(x,z)^q$ is also a valid kernal.

10.2 (a)

① We will first show 10.1 (a)

$k(x,z) = ck_1(x,z)$ is a valid kernal.

$$k(x,z) = [\sqrt{c}\,\phi'_1(x), \ldots, \sqrt{c}\,\phi'_{v_1}(x)]^T [\sqrt{c}\,\phi'_1(x), \ldots, \sqrt{c}\,\phi'_{v_1}(x)]$$

$$= \alpha k_1(x,z)$$

② We then show 10.1 (b)

Define $\phi(x) = [\phi^1_1(x), \ldots, \phi^1_{v_1}(x), \phi^2_1(x), \ldots, \phi^2_{v_2}(x)]$

$$\Rightarrow \phi^T(x)\cdot\phi(z) = \phi^1_1(x)\cdot\phi^1_1(z) + \cdots + \phi^1_{v_1}(x)\cdot\phi^1_{v_1}(z) +$$

$$\phi^2_1(x)\,\phi^2_1(z) + \cdots + \phi^2_{v_2}(x)\,\phi^2_{v_2}(z)$$

$$= \phi^1(x)^T\phi^1(z) + \phi^2(x)^T\phi^2(z)$$

$$= k_1(x,z) + k_2(x,z)$$

③ We then show 10.1 (f)

$k(x,z) = \exp(k_1(x,z))$ is a valid kernel

$$\exp(k_1(x,z))$$

$$= \exp(0) + \exp(0)'\,k_1(x,z) + \frac{\exp(0)''}{2!}(k_1(x,z))^2 + \cdots$$

(Taylor expansion)

$$= 1 + k_1(x,z) + \frac{1}{2}(k_1(x,z))^2 + \cdots$$

(Using the conclusions we draw above, $\exp(k_1(x,z))$ is a valid kernel if $k_1(x,z)$ is valid)

We now show $k(x,z) = \exp(-\alpha\|x-z\|^2)$, $\alpha > 0$ is valid kernel

$$k(x,z) = \exp(-\alpha\|x\|^2)\exp(-\alpha\|z\|^2)\exp(2\alpha x^T z)$$

$k_1(x,z) = [x_1^2, x_2^2, \dots, x_n^2]^T [1, 1, \dots, 1]$ valid

$k_2(x,z) = [1, 1, \dots, 1]^T [z_1^2, \dots, z_n^2]$ valid

$k_3(x,z)$ is linear kernal

$\Rightarrow$ By 10.1(a)

$-\alpha k_1(x,z)$, $-\alpha k_2(x,z)$, $2\alpha k_3(x,z)$ are valid

By 10.1(f)

$\exp(-\alpha k_1(x,z))$, $\exp(-\alpha k_2(x,z))$, $\exp(2\alpha k_3(x,z))$ valid

By 10.1(c)

$\exp(-\alpha k_1(x,z)) \times \exp(-\alpha k_2(x,z)) \times \exp(2\alpha k_3(x,z))$ is valid

Hence $k(x,z)$ is a valid kernal

10.4

(a)

$$\tilde{k}(x,z) = \frac{k(x,z)}{\sqrt{\Phi^T(x)\Phi(x)\ \Phi^T(z)\Phi(z)}}$$

$$= f(x)\cdot k(x,z) f(z)$$

where $f(x), f(z)$ are some scalar functions

$f(x) = \sqrt{\Phi^T(x)\Phi(x)}$

By 10.1(d)

$k(x,z) = f(x)\cdot k_1(x,z) f(z)$ is valid kernel.

proof: Consider $\phi(x) = [f(x)\cdot\phi_1'(x), \ldots\ldots, f(x)\phi_n'(x)]$

$$k(x,z) = \phi(x)^T\phi(z) = f(x)\phi_1'(x)\phi_1'(z)f(z) + \cdots\cdots$$

$$= f(x)\cdot k_1(x,z)\cdot f(z)$$

Hence $\Rightarrow$ $\tilde{k}(x,z)$ is also a valid kernel

(b) $$\tilde{k}(x,z) = \frac{\Phi(x)^T\Phi(z)}{\sqrt{\Phi(x)^T\Phi(z)\cdot\Phi(z)^T\Phi(z)}}$$

$$= \frac{\Phi(x)\bullet\Phi(z)}{|\Phi(x)|\cdot|\Phi(z)|}$$

which is the definition of cos in high-dim space.

(c) By Cauchy's inequality

$$\left((\sum_i a_i)(\sum_i b_i)\right)^2 \le \sum_i a_i^2 \cdot \sum_i b_i^2$$

Hen

$$\left((\sum_i \Phi(x)_i)(\sum_i \Phi(z)_i)\right)^2 \le \sum_i (\Phi(x)_i)^2 \sum_i (\Phi(z)_i)^2$$

$$\Rightarrow \quad -1 \le \frac{\Phi(x)\cdot\Phi(z)}{|\Phi(x)||\Phi(z)|} \le 1$$

10.10

(a) Primal:

$$l(w,b,\alpha) = \frac{1}{2}w^Tw - \sum_i \alpha_i(y_i(w^T\Phi(x_i)+b)-1)$$

$$\frac{\partial l}{\partial w} = w - \sum_i \alpha_i y_i \Phi(x_i) = 0$$

$$\Rightarrow \quad w^* = \sum \alpha_i y_i \Phi(x_i)$$

$$b^* = \frac{1}{|SV|} \sum_{i \in SV} \left(y_i - \sum_j \alpha_j y_j \Phi(x_i)^T \Phi(x_j) \right)$$
$$= \frac{1}{|SV|} \sum_{i \in SV} \left(y_i - \sum_j \alpha_j y_j k(x_i, x_j) \right)$$

10.11

(a) $$w = (X' R X'^T + P)^{-1} X' R z$$

where $X' = [\Phi(x_1), \ldots, \Phi(x_n)]$

By $(P^{-1} + B^T R^{-1} B)^{-1} B^T R^{-1} = P B^T (B P B^T + R)^{-1}$

$\alpha_* = w^T \Phi(x_*) = \Phi(x_*)^T w = \Phi(x_*)^T P^{-1} X' (X'^T P^{-1} X' + R^{-1})^{-1} z$

$\underbrace{\text{since } P^{-1} \text{ is symmetric}}$

We can define $k(x_i, x_j) = \Phi(x_i)^T P^{-1} \Phi(x_j)$

$$\Rightarrow \alpha_* = k_*^T (K + R^{-1})^{-1} z$$

(b) $$\alpha^{(old)} = X'^T w$$
$$= X'^T P^{-1} X' (X'^T P^{-1} X' + R^{-1})^{-1} z$$
$$= K (K + R^{-1})^{-1} z$$

$$z = \alpha^{(old)} - R^{-1} (\pi - y)$$

(c) One interpretation is that parameter in P is embedded in the new kernel (prior)

(d)

The "kernel scale parameter" is called "gamma" in LibSVM. Consider the Gaussian kernel: k(x,y) = exp (-gamma * (x-y)^2). If gamma is large, then this kernel will fall off rapidly as the point y moves away from x. As gamma decreases, the kernel will fall off less and less rapidly. When gamma is 0, the kernel will be the same (=1) for all points y irrespective of where y is in the feature space.

In this interpretation, gamma is related to how spread out your data points are. If they are very far from each other (which would happen in a very high dimensional space for example), then you don't want the kernel to drop off quickly, so you would use a small gamma. Thus libSVM uses a default of 1/num_features.

As for how to set it, the answer will have to be cross-validation.

10.13

(a) No, it does not matter, since it is the direction of w matters, but learning rate just scale its magnitude.

(b) Iteratively

$$w = \sum_{i=1}^{n} \eta y_i x_i \cdot k_i$$

where k_i is the total number of times when x_i is misclassified.

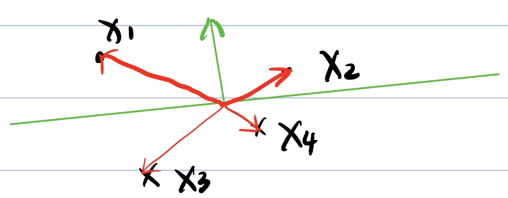
$$= \sum_{i}^{n} \alpha_i y_i x_i$$

(c) originally, each time when it is misclassified, we add $y_i x_i$ scaled by η for w and $y_i R^2$ scaled by η for b.

So just add them scaled by 1

Finally, it will converges since the final result are same form.

(d)



α_i : weight for the vector for each data points
where its combination determines the direction of
the line

$\alpha_i y_i + \alpha_j y_j \approx 0 \Rightarrow$ hard to classify.

(e) $$y_* = \text{sign}(w^T \Phi(x_*) + b)$$
$$= \text{sign}\left(\sum_i \alpha_i y_i \Phi(x_i)^T \Phi(x_*) + b\right)$$
$$= \text{sign}\left(\sum_i \alpha_i y_i k(x_*, x_i) + b\right)$$

10.14

(a) $$d(x, \mu_k) = \|x - \mu_k\|^2$$
$$= (x-\mu_k)^T(x-\mu_k)$$
$$= x^T x - 2x^T \mu_k + \mu_k^T \mu_k$$
$$= x^T x - 2\frac{1}{N_k}\sum_{l=1}^{n} z_{lk} x^T x_l + \frac{1}{N_k^2}\sum_l \sum_m z_{lk} z_{mk} x_l^T x_m$$

(b) Just substitute

(c) when $\frac{1}{N_k}\sum_l z_{lk} \cdot k(x, x_l)$ is as large as possible.
i.e. for those assigned to cluster j
maximize $\sum_{z \in n_k} e^{-a(x-x_l)^2}$
put x at a $\underbrace{\text{high density}}_{\text{most } x_i \in n_k \text{ locate at.}}$ region

PS 10.15

(a) If it is not valid, then there must be a dimension
which is orthogonal to the space spanned by X.
If we project w_{opt} to the spanned space.

The effect is equivalent.

(b) $\mu_j = \frac{1}{n_j} [x_{j1}, x_{j2}, \dots, x_{jn_j}] \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$

$= \frac{\sum_i^{n_j} x_{ji}}{n_j}$

$S_j = \sum_{i \in J} (x_i - \mu_j)(x_i - \mu_j)^T$

$= \sum_{i \in J} x_i x_i^T - n_j \mu_j \mu_j^T$

$= X_j X_j^T - \frac{1}{n_j} X_j 1 1^T X_j^T$

(c) $w^T \mu_j = \alpha^T X^T \frac{1}{n_j} X_j 1$

$= \alpha^T \hat{\mu}_j$ $\quad \begin{cases} \hat{\mu}_j = X^T \frac{1}{n_j} X_j 1 \\ \hat{S}_j = X^T X_j (I - \frac{1}{n_j} 1 1^T) X_j^T X \end{cases}$

$w^T S_j w = \alpha^T \hat{S}_j \alpha$

(d) directly substitute

(e) $w^T S_B w = w^T (\mu_i - \mu_j)(\mu_i - \mu_j)^T w$

$= (\alpha^T \hat{\mu}_i - \alpha^T \hat{\mu}_j)(\alpha^T \hat{\mu}_i - \alpha^T \hat{\mu}_j)^T$

$= \alpha^T \hat{S}_B \alpha$

$w^T S_W w = w^T (S_i + S_j) w$

$= \alpha^T (\hat{S}_i + \hat{S}_j) \alpha$

$= \alpha^T \hat{S}_W \alpha$

(f) $\alpha^* = \arg\max_\alpha J(\alpha)$

$J(\alpha) = \frac{\alpha^T \hat{S}_B \alpha}{{}^T \hat{S}_W \alpha}$

Constrain $\alpha^T \hat{S}_W \alpha = 1$

$\Rightarrow l(\alpha) = \alpha^T \hat{S}_B \alpha - \lambda (\alpha^T \hat{S}_W \alpha - 1) = 0$

$\frac{\partial l}{\partial \alpha} = 0$

$\Rightarrow \hat{S}_B \alpha - \lambda \hat{S}_W \alpha = 0$

$(\hat{\mu}_2 - \hat{\mu}_1)(\hat{\mu}_2 - \hat{\mu}_1)^T \alpha = \lambda \hat{S}_W \alpha$

$\alpha \propto \hat{S_w}^{-1}(\hat{\mu}_2 - \hat{\mu}_1)$

(g) $z = w^T x$

$= \alpha^T X^T \cdot x$

$= \sum_i \alpha_i k(x_i, x)$