CS5487 Problem Set

Solutions - Homework and Tutorials

Antoni Chan
Department of Computer Science
City University of Hong Kong

Important Note: These problem set solutions are meant to be a study aid for the final exam *only*. They should not be used as "model answers" to help do the problem set. The point of the problem set is to encourage students to think critically about machine learning problems – part of this is to learn to check your own answers, examine them for flaws, and be confident that they are correct. This is an important skill for PhD students to obtain, since research, by definition, involves solving unsolved problems. There are no "model answers" in research, so its best to get used to not having them available.

For current and former CS5487 students:

- DO NOT give the solutions to current CS5487 students.
- DO NOT post electronic copies of the solutions online.
- DO NOT use the solutions as "model answers" for doing the problem set.
- DO use these solutions as a study aid for the final exam.

version v1: 4220e5da9c68ecfd5d8536fe34e0bcfb

Homework Problems

Problem 2.1 The Poisson distribution and flying bombs

(a) The log-likelihood of the data $\mathcal{D} = \{k_1, \cdots, k_N\}$ is

$$\log p(\mathcal{D}) = \sum_{i=1}^{N} [-\lambda + k_i \log \lambda - \log k_i!] = -N\lambda + (\sum_{i=1}^{N} k_i) \log \lambda - \sum_{i=1}^{N} \log k_i! \tag{S.1}$$

We obtain the maximum by setting the derivative with respect to λ to 0,

$$\frac{\partial \log p(\mathcal{D})}{\partial \lambda} = -N + \frac{1}{\lambda} \sum_{i=1}^{N} k_i = 0 \quad \Rightarrow \quad \hat{\lambda} = \frac{1}{N} \sum_{i=1}^{N} k_i, \tag{S.2}$$

which is the sample average of the counts. The constraint that $\lambda \geq 0$ is naturally satisfied since $k_i \geq 0$. Looking at the second derivative,

$$\frac{\partial^2 \log p(\mathcal{D})}{\partial \lambda^2} = -\frac{1}{\lambda^2} \sum_{i=1}^{N} k_i, \tag{S.3}$$

it is always negative (or zero in the degenerate case), and hence we indeed have a maximum of log-likelihood.

(b) Let x_i be the sample random variables, where each x_i is distributed as a Poisson with parameter λ. The mean of the estimator using samples $X = \{x_1, \cdots, x_N\}$ is

$$\mathbb{E}_X[\hat{\lambda}] = \mathbb{E}_X\left[\frac{1}{N}\sum_{i=1}^{N} x_i\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_X[x_i] = \frac{1}{N}\sum_{i=1}^{N}\lambda = \lambda. \tag{S.4}$$

$$\Rightarrow \quad \mathbb{E}_X[\hat{\lambda} - \lambda] = 0. \tag{S.5}$$

Hence the ML estimate $\hat{\lambda}$ is unbiased. For the variance, we have

$$\mathrm{var}_X[\hat{\lambda}] = \mathbb{E}_X\left[(\hat{\lambda} - \mathbb{E}_X[\hat{\lambda}])^2\right] = \mathbb{E}_X\left[\left(\frac{1}{N}(\sum_{i=1}^{N} x_i) - \lambda\right)^2\right] = \mathbb{E}_X\left[\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \lambda)\right)^2\right] \tag{S.6}$$

$$= \frac{1}{N^2}\mathbb{E}_X\left[\sum_{i=1}^{N}\sum_{j=1}^{N}(x_i - \lambda)(x_j - \lambda)\right] = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbb{E}_X[(x_i - \lambda)(x_j - \lambda)] \tag{S.7}$$

$$= \frac{1}{N^2}\left(\sum_{i=j}\mathbb{E}_X[(x_i - \lambda)^2] + \sum_{i\neq j}\mathbb{E}_X[(x_i - \lambda)(x_j - \lambda)]\right) \tag{S.8}$$

For $i \neq j$, we have

$$\mathbb{E}_{x_i,x_j}[(x_i - \lambda)(x_j - \lambda)] = \mathbb{E}_{x_i}[x_i - \lambda]\mathbb{E}_{x_j}[x_j - \lambda] = 0, \tag{S.9}$$

which uses the independence assumption. For $i = j$, we have

$$\mathbb{E}_{x_i}[(x_i - \lambda)^2] = \mathrm{var}(x_i) = \lambda. \tag{S.10}$$

Substituting into (S.8),

$$\mathrm{var}_X[\hat{\lambda}] = \frac{1}{N^2}(N\lambda) = \frac{\lambda}{N}. \tag{S.11}$$

Hence the variance of the estimator is inversely proportional to the number of samples.

(c) The number of cells is 576. Assuming that the observation for the "5 and over" bin was actually 5, the ML estimate for the Poisson distribution is

$$\hat{\lambda} = \frac{1}{576}(229(0) + 211(1) + 93(2) + 35(3) + 7(4) + 5(1)) = 0.929. \tag{S.12}$$

Note that increasing the value of the "5 and over" bin to 6 slightly increases the estimate by $1/576 = 0.0017$.

(d) Using $\hat{\lambda}$ from (c), the expected counts for 576 observations is given in the table below.

k	0	1	2	3	4	5+
$p(x = k)$	0.3950	0.3669	0.1704	0.0528	0.0122	0.0027
expected	227.5	211.3	98.1	30.4	7.1	1.5
observed	229	211	93	35	7	1

The expected and observed counts are very similar. For $k = 2$ and $k = 3$ the expected and observed counts are off by around 5, but other values are very close. Because the expected and observed data match well, this supports the conclusion that the flying bombs were falling randomly, and not due to any precision targeting. Quantitatively, a χ^2 test can be used to check the goodness-of-fit between the expected and actual data. In this case, the χ^2 test gives a high probability (0.888) that the observations came from the expected distribution.

.....

Problem 2.10 Robust Regression and MLE

(a) Assuming that the observation noise is Laplace distribution will yield the robust regression formulation. Formally, we have a function $f(x) = \phi(x)^T\theta$. We observe noisy outputs of $f(x)$,

$$y_i = f(x_i) + \epsilon_i, \tag{S.13}$$

where ϵ_i is distributed as a Laplace distribution with zero-mean,

$$p(\epsilon_i) = \frac{1}{2\lambda} e^{-\frac{|\epsilon_i|}{\lambda}}. \tag{S.14}$$

Hence, y_i is also a Laplace distribution, but with mean $f(x_i)$,

$$p(y_i|x_i) = \frac{1}{2\lambda} e^{-\frac{|y_i - f(x_i)|}{\lambda}}. \tag{S.15}$$

Taking the log and summing over the training points, we get the data log-likelihood

$$\log p(y|X) = -\frac{1}{\lambda} \sum_{i=1}^{n} |y_i - \phi(x_i)^T\theta|. \tag{S.16}$$

Since λ is a constant that does not affect the optimization, the MLE solution is then

$$\theta^* = \operatorname*{argmax}_\theta \log p(y|X) = \operatorname*{argmax}_\theta - \sum_{i=1}^{n} |y_i - \phi(x_i)^T\theta| = \operatorname*{argmin}_\theta \sum_{i=1}^{n} |y_i - \phi(x_i)^T\theta| \tag{S.17}$$

In summary, robust regression (minimizing the L1 norm) is equivalent to assuming a Laplace distribution on the observation noise and applying MLE.

The L1 norm is more robust to outliers because it gives less penalty to large errors than the L2 norm. This is illustrated in Figure 1a. The L2 norm gives a squared penalty to large errors, which in turn makes the regression algorithm focus more on reducing those errors. In contrast, the L1 norm gives less penalty, thus allowing the regression to focus less on large errors. Another way to think about it is to consider the derivatives of the L2 and L1 error functions. For L2, the error is decreased the most by reducing the largest error. For L1, the error decreased equally for regardless of the amount of error.

A second way to explain how L1 is more robust to outliers is to consider the associated probabilistic model. Noting that outliers are rare events of large amplitude, the ability of a given model to explain outliers is a function of how much probability mass is contained in its tails. In particular, an *heavy-tailed* distribution (more probability mass in the tails) will explain outliers better than a distribution that is not heavy-tailed. It is indeed the case that the Laplacian is

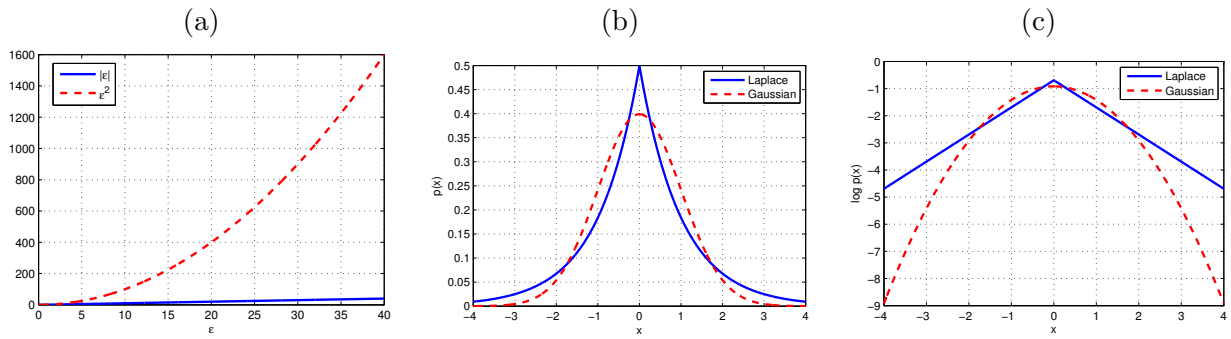

Figure 1: (a) comparison of L1 and L2 norms; (b) Gaussian and Laplace distributions; (c) log distributions.

more heavy-tailed than the Gaussian. You can convince yourself of this by plotting the two distributions. Notice that, since both functions are exponentially decreasing with the distance from the mean, it is usually difficult to see anything about the tails in the plot of the pdf. A better strategy is to look at the plot of the log of the pdf, which makes the differences more salient. Both are shown in Figure 1b and 1c, for the Gaussian and Laplacian distributions when $\sigma^2 = 1$. Figure 1b shows the pdf plots, and Figure 1c shows their log. As you can see, the Gaussian decays much faster (its log is a quadratic function of the distance from the mean) than the Laplacian (log is a linear function of this distance). For example, for x five standard deviations away from the mean, the difference between the two functions is about 8 dbs, i.e. the Laplace probability is about $3,000$ larger. This implies that Laplacian noise explains outliers much better than Gaussian noise. Hence, in the presence of outliers there will be less of a mismatch under the Laplacian model and the ML solution is therefore better than that achievable with the Gaussian.

(b) This is a optimization trick to turn an absolute value in the objective function into inequality constraints. t_i is an upper bound on each term $|y_i - \phi(x_i)^T\theta|$. At the optimum, we must have $t_i = |y_i - \phi(x_i)^T\theta|$. Otherwise, we wouldn't be at a minimum (we could still decrease t_i and hence decrease the objective function). Hence, at the optimum of (2.16), the minimum of the original objective (2.15) has been found.

(c) Note that an inequality constraint $|a| \leq b$ can be turned into two inequality constraints,

$$|a| \leq b \quad \Rightarrow \quad -b \leq a \leq b \quad \Rightarrow \quad \begin{cases} a \leq b \\ -a \leq b \end{cases} \tag{S.18}$$

If a is positive, then the first inequality bounds it. If a is negative, then the second inequality bounds its magnitude by b. Hence, the ith inequality constraint in (2.16) can be written as

$$y_i - \phi(x_i)^T\theta \leq t_i, \tag{S.19}$$

$$-(y_i - \phi(x_i)^T\theta) \leq t_i, \tag{S.20}$$

or equivalently

$$-\phi(x_i)^T\theta - t_i \leq -y_i \tag{S.21}$$

$$\phi(x_i)^T\theta - t_i \leq y_i. \tag{S.22}$$

Stacking the inequalities into a vector, we obtain the definitions of $\mathbf{A}$ and $\mathbf{b}$ in (2.18). Setting $\mathbf{f} = \begin{bmatrix} 0_D \\ 1_n \end{bmatrix}$ will sum over the t_i's.

. . . . . .

Problem 3.7 Bayesian estimation for a Bernoulli distribution

(a) Assuming independent samples,

$$p(\mathcal{D}|\pi) = \prod_{i=1}^{n} p(x_i|\pi) = \prod_{i=1}^{n} \pi^{x_i}(1-\pi)^{1-x_i} = \pi^{\sum_{i=1}^n x_i}(1-\pi)^{\sum_{i=1}^n (1-x_i)} \tag{S.23}$$

$$= \pi^s(1-\pi)^{n-s}, \quad s = \sum_{i=1}^{n} x_i. \tag{S.24}$$

(b) Assuming a uniform prior $p(\pi) = 1$, and applying Bayes' rule,

$$p(\pi|\mathcal{D}) = \frac{p(\mathcal{D}|\pi)p(\pi)}{\int_0^1 p(\mathcal{D}|\pi)p(\pi)d\pi} = \frac{p(\mathcal{D}|\pi)}{\int_0^1 p(\mathcal{D}|\pi)d\pi} = \frac{\pi^s(1-\pi)^{n-s}}{\int_0^1 \pi^s(1-\pi)^{n-s}d\pi} \tag{S.25}$$

$$= \frac{\pi^s(1-\pi)^{n-s}}{\frac{s!(n-s)!}{(s+n-s+1)!}} = \frac{(n+1)!}{s!(n-s)!}\pi^s(1-\pi)^{n-s}, \tag{S.26}$$

which uses (3.31).

For $n = 1$ and $s \in \{0, 1\}$, the posterior is

$$p(\pi|\mathcal{D}) = 2\pi^s(1-\pi)^{1-s}. \tag{S.27}$$

The plots of the two densities are shown in Figure 2. Notice that the observation of a "0" turns the uniform prior into a posterior with a lot more probability for small values of π (which is the probability of "1"). On the other hand, the observation of a "1" turns the uniform prior into a posterior with a lot more probability for large values of π. As usual for Bayesian inference, this is intuitive.

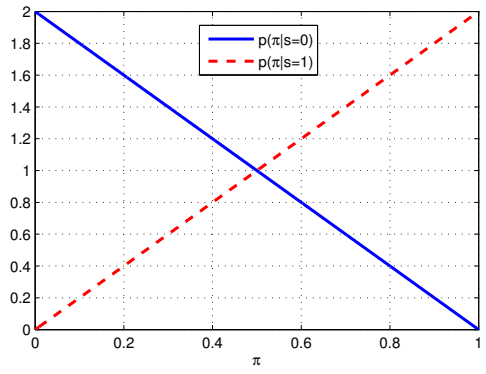

Figure 2: Posteriors for π.

(c) The predictive distribution is obtained by integrating the observation likelihood with the posterior,

$$p(x|\mathcal{D}) = \int_0^1 p(x|\pi)p(\pi|\mathcal{D})d\pi = \int_0^1 \pi^x(1-\pi)^{1-x}\frac{(n+1)!}{s!(n-s)!}\pi^s(1-\pi)^{n-s}d\pi \tag{S.28}$$

$$= \frac{(n+1)!}{s!(n-s)!}\int_0^1 \pi^{s+x}(1-\pi)^{n-s+1-x}d\pi = \frac{(n+1)!}{s!(n-s)!}\frac{(s+x)!(n-s+1-x)!}{(n+2)!} \tag{S.29}$$

$$= \frac{(n+1)!}{(n+2)!}\frac{(s+x)!}{s!}\frac{(n-s+1-x)!}{(n-s)!} = \frac{1}{n+2}(s+1)^x(n-s+1)^{1-x} \tag{S.30}$$

$$= \left(\frac{s+1}{n+2}\right)^x\left(\frac{n-s+1}{n+2}\right)^{1-x} = \left(\frac{s+1}{n+2}\right)^x\left(1-\frac{s+1}{n+2}\right)^{1-x}, \tag{S.31}$$

where (S.29) follows from (3.31) and (S.30) uses $x \in \{0, 1\}$. The effective estimate for $\pi = \frac{s+1}{n+2}$, which can be explained as adding two "virtual" samples, "0" and "1", to the MLE estimate.

(d) The ML solution is

$$\pi_{ML} = \operatorname*{argmax}_\pi \log p(\mathcal{D}|\pi) = \operatorname*{argmax}_\pi s\log\pi + (n-s)\log(1-\pi) \tag{S.32}$$

Note that we should impose the constraints $0 \leq \pi \leq 1$. However, as we will see below, the maximum of the unconstrained problem is inside the constraint region. Hence, we can get away without doing this. It therefore suffices to compute the derivative of $\ell = \log p(\mathcal{D}|\pi)$,

$$\frac{\partial\ell}{\partial\pi} = \frac{s}{\pi} - \frac{n-s}{1-\pi} \tag{S.33}$$

and set to zero, from which we obtain

$$\frac{s}{\pi} - \frac{n-s}{1-\pi} = 0 \tag{S.34}$$

$$s(1-\pi) - (n-s)\pi = 0, \tag{S.35}$$

$$\Rightarrow \pi^* = \frac{s}{n}. \tag{S.36}$$

Since $s \leq n$ this is indeed inside the constraint region. However, we still need to check that we have a maximum. For this we compute the 2nd derivative

$$\frac{\partial^2\ell}{\partial\pi^2} = -\frac{s}{\pi^2} - \frac{n-s}{(1-\pi)^2}. \tag{S.37}$$

This is always negative, and hence we have a maximum. Furthermore the function $\ell(\pi)$ is concave and this maximum is global. Hence, we do not need to check boundary constraints.

The MAP solution is

$$\hat{\pi} = \operatorname*{argmax}_\pi p(\pi|\mathcal{D}) = \operatorname*{argmax}_\pi \log p(\mathcal{D}|\pi) + \log p(\pi) \tag{S.38}$$

Notice that, since $p(\pi)$ is uniform, the second term does not depend on π and can be dropped. Hence, the MAP solution is the same as the ML solution. This result is always true when we have a uniform prior, in which case there is no point in favoring MAP over ML.

(e) For the first prior, $p_1(\pi) = 2\pi$, substituting into (S.38),

$$\ell_1 = \log p(\mathcal{D}|\pi) + \log p(\pi) = s \log \pi + (n-s)\log(1-\pi) + \log 2\pi, \tag{S.39}$$

$$\frac{\partial \ell_1}{\partial \pi} = \frac{s}{\pi} - \frac{n-s}{1-\pi} + \frac{1}{\pi} = 0 \tag{S.40}$$

$$\Rightarrow s(1-\pi) - (n-s)\pi + (1-\pi) = 0 \tag{S.41}$$

$$\Rightarrow \hat{\pi}_1 = \frac{s+1}{n+1}. \tag{S.42}$$

For the second prior, $p_0(\pi) = 2 - 2\pi$,

$$\ell_0 = \log p(\mathcal{D}|\pi) + \log p(\pi) = s \log \pi + (n-s)\log(1-\pi) + \log 2(1-\pi) \tag{S.43}$$

$$\frac{\partial \ell_0}{\partial \pi} = \frac{s}{\pi} - \frac{n-s}{1-\pi} - \frac{1}{1-\pi} = 0 \tag{S.44}$$

$$\Rightarrow s(1-\pi) - (n-s)\pi - \pi = 0 \tag{S.45}$$

$$\Rightarrow \hat{\pi}_0 = \frac{s}{n+1}. \tag{S.46}$$

For these two priors, it is equivalent to adding a virtual sample to the training set, a "1" for $p_1(\pi)$ and a "0" for $p_0(\pi)$. These virtual samples reflect the bias of the prior.

In summary, for all these cases, the predictive distribution (the distribution of a new sample after learning the parameter π) is of the form

$$p(x|\mathcal{D}) = \hat{\pi}^x (1-\hat{\pi})^{1-x} \tag{S.47}$$

where $\hat{\pi}$ is determined by the estimation method,

Estimator	$\hat{\pi}$	# tosses	# "1"s	interpretation
ML	s/n	n	s	
MAP (non-informative)	s/n	n	s	same as ML
MAP (favor 1s)	$(s+1)/(n+1)$	$n+1$	$s+1$	add an extra "1"
MAP (favor 0s)	$s/(n+1)$	$n+1$	s	add an extra "0"
Bayesian (non-informative)	$(s+1)/(n+2)$	$n+2$	$s+1$	add one of each.

In all cases, they estimate can be interpreted as an ML estimate with extra sample(s) added to reflect the bias of the prior.

. . . . . .

Problem 4.5 Flying Bombs, part II – EM for mixtures of Poissons

(a) Define z_i as the hidden assignment variable that assigns sample x_i to mixture component $z_i = j$. The complete data likelihood is

$$p(X, Z) = \prod_{i=1}^{n} p(z_i) p(x_i|z_i) = \prod_{i=1}^{n} \pi_{z_i} p(x_i|z_i). \tag{S.48}$$

Using the indicator variable trick, by defining $z_{ij} = 1$ iff $z_i = j$ and 0 otherwise, we have

$$p(X, Z) = \prod_{i=1}^{n} \prod_{j=1}^{K} \pi_j^{z_{ij}} p(x_i|z_i = j)^{z_{ij}}, \tag{S.49}$$

and taking the log,

$$\log p(X, Z) = \sum_{i=1}^{n} \sum_{j=1}^{K} z_{ij} \log \pi_j + z_{ij} \log p(x_i | z_i = j). \tag{S.50}$$

For the E-step, we obtain the Q function by taking the expectation of the complete data log-likelihood in (S.50),

$$\begin{aligned} Q(\theta; \hat{\theta}) &= \mathbb{E}_{Z|X,\hat{\theta}} \left[\log p(X, Z|\theta) \right] && \text{(S.51)} \\ &= \mathbb{E}_{Z|X,\hat{\theta}} \left[\sum_{i=1}^{n} \sum_{j=1}^{K} z_{ij} \log \pi_j + z_{ij} \log p(x_i | z_i = j) \right] && \text{(S.52)} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{K} \hat{z}_{ij} \log \pi_j + \hat{z}_{ij} \log p(x_i | z_i = j), && \text{(S.53)} \end{aligned}$$

where the last line follows because the expectation only applies to variable z_{ij}. The "soft assignment" term $\hat{z}_{ij}$ is calculated using the current parameter estimates $\hat{\theta}$,

$$\begin{aligned} \hat{z}_{ij} &= \mathbb{E}_{Z|X,\hat{\theta}}[z_{ij}] = p(z_i = j | X, \hat{\theta}) && \text{(S.54)} \\ &= \frac{p(X|z_i = j) p(z_i = j)}{p(X)} = \frac{p(X_{\neg i}) p(x_i|z_i = j) p(z_i = j)}{p(X_{\neg i}) p(x_i)} && \text{(S.55)} \\ &= \frac{\pi_j p(x_i|z_i = j)}{\sum_{k=1}^{K} \pi_k p(x_i|z_i = k)} = p(z_i = j | x_i, \hat{\theta}). && \text{(S.56)} \end{aligned}$$

(S.55) follows from the independence assumption of the samples, and $X_{\neg i}$ is the set of x_k with $k \neq i$. Note that we have not used any properties of the mixture components yet, so this is the general form of the Q function for any mixture model.

For the M-step, we maximize Q with respect to the parameters θ. First, we optimize the component priors. We have a constraint that $\sum_j \pi_j = 1$, and $\pi_j \geq 0$. For the equality constraint (the non-negative constraint is naturally satisfied), define the Lagrangian as

$$L(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{K} \hat{z}_{ij} \log \pi_j + \lambda (1 - \sum_{j=1}^{K} \pi_j), \tag{S.57}$$

where λ is the Lagrange multiplier. Taking the derivatives and setting to 0,

$$\frac{\partial L}{\partial \lambda} = 1 - \sum_{j=1}^{K} \pi_j = 0 \quad \Rightarrow \quad \sum_{j=1}^{K} \pi_j = 1, \tag{S.58}$$

$$\frac{\partial L}{\partial \pi_j} = \sum_{i=1}^{n} \frac{\hat{z}_{ij}}{\pi_j} + \lambda = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \hat{z}_{ij} + \lambda \pi_j = 0. \tag{S.59}$$

Summing (S.59) over j, we have

$$\sum_{j=1}^{K} \sum_{i=1}^{n} \hat{z}_{ij} + \lambda \sum_{j=1}^{K} \pi_j = 0 \quad \Rightarrow \quad \lambda = -n, \tag{S.60}$$

which follows from $\sum_j \hat{z}_{ij} = 1$ and $\sum_j \pi_j = 1$. Finally, substituting into (S.59), we have

$$\hat{\pi}_j = \frac{1}{n} \sum_{i=1}^{n} \hat{z}_{ij} = \frac{\hat{n}_j}{n}, \tag{S.61}$$

where $\hat{n}_j = \sum_{i=1}^{n} \hat{z}_{ij}$ is the (soft) number of samples assigned to component j. Again this is a standard result for any mixture model.

Next, we look at optimizing λ_j for each Poisson component. Note that the parameters of the jth component do not affect the other components, so each λ_j can be optimized separately, by maximizing

$$\ell_j = \sum_{i=1}^{n} \hat{z}_{ij} \log p(x_i | z_i = j) \tag{S.62}$$

$$= \sum_{i=1}^{n} \hat{z}_{ij} (-\lambda_j + x_i \log \lambda_j - \log x_i!) \tag{S.63}$$

$$= -\lambda_j (\sum_{i=1}^{n} \hat{z}_{ij}) + (\sum_{i=1}^{n} \hat{z}_{ij} x_i) \log \lambda_j - \sum_{i=1}^{n} \hat{z}_{ij} \log x_i! \tag{S.64}$$

Taking the derivative and setting to zero,

$$\frac{\partial \ell_j}{\partial \lambda_j} = -\sum_{i=1}^{n} \hat{z}_{ij} + \frac{1}{\lambda_j} \sum_{i=1}^{n} \hat{z}_{ij} x_i = 0 \tag{S.65}$$

$$\Rightarrow \hat{\lambda}_j = \frac{1}{\sum_{i=1}^{n} \hat{z}_{ij}} \sum_{i=1}^{n} \hat{z}_{ij} x_i = \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{z}_{ij} x_i \tag{S.66}$$

In summary, the EM algorithm for mixture of Poissons is

$$\mathrm{E-step}: \ \hat{z}_{ij} = p(z_i = j | x_i, \hat{\theta}) = \frac{\hat{\pi}_j p(x_i | z_i = j, \hat{\lambda}_j)}{\sum_{k=1}^{K} \hat{\pi}_k p(x_i | z_i = k, \hat{\lambda}_j)} \tag{S.67}$$

$$\mathrm{M-step}: \ \hat{n}_j = \sum_{i=1}^{n} \hat{z}_{ij}, \quad \hat{\pi}_j = \frac{\hat{n}_j}{n}, \quad \hat{\lambda}_j = \frac{1}{\hat{n}_j} \sum_{i=1}^{n} \hat{z}_{ij} x_i. \tag{S.68}$$

The estimate of λ_j, the arrival rate, is similar to the standard ML estimate, except a weighted average is used with EM, where the weights are the posterior probabilities of the sample being assigned to component j. Note that for any exponential family distribution, we will get a similar form in the M-step, which is a weighted average of the sufficient statistics.

(b) A mixture of Poissons was learned from the data of each city with number of components $K \in \{1, 2, 3, 4\}$. The final data log-likelihood and model parameters are in the following table, and the mixture distributions are plotted in Figure 3 (top).

city	K	loglike	(π_1, λ_1)	(π_2, λ_2)	(π_3, λ_3)	(π_4, λ_4)
London	1	-728.713	(1, 0.929)			
	2	-728.703	(0.736, 0.988)	(0.264, 0.765)		
	3	-728.701	(0.599, 0.982)	(0.206, 0.972)	(0.195 0.721)	
	4	-728.703	(0.099, 0.995)	(0.317, 0.989)	(0.330, 0.981)	(0.254, 0.761)
Antwerp	1	-830.696	(1, 0.896)			
	2	-748.025	(0.339, 2.195)	(0.661, 0.230)		
	3	-747.665	(0.278, 2.374)	(0.451, 0.523)	(0.271, 0.0001)	
	4	-747.665	(0.278, 2.374)	(0.234, 0.523)	(0.217, 0.523)	(0.271, 0.0001)

There are a few interesting observations. First, for the London data, the log-likelihood does not increase significantly as K increases. This suggests that the London data can already be explained well with one component. Looking at the model parameters for $K = 2$, there seem to be two important values of λ, 0.98 and 0.76. The first is the dominant value (larger prior weight), while the second provides some correction for the number of samples with count 2 and 3 (see Figure 3 top-left). While EM has found this second component, the fact that the log-likelihood is similar to $K = 1$ indicates that the variations in the sample data are equally well explained by the single component. In any case, 0.98 and 0.76 are relatively similar values. For larger values of K, the extra components have values similar to the original 0.98.

Looking at the Antwerp models, there is a clear increase in log-likelihood as K increases, until $K = 3$. For $K = 1$, the value of λ is 0.896, but looking at the plot in Figure 3 (top-right) shows that the single Poisson does not model the data well. Increasing the number of components to $K = 2$, we recover two significant components, $\lambda_1 = 2.195$ and $\lambda_2 = 0.230$. The difference between the two values suggest that some squares are hit more often than others. Further increasing to $K = 3$ recovers three types of squares: 1) frequently hit ($\lambda_1 = 2.374$), 2) infrequently hit ($\lambda_2 = 0.523$), and 3) never hit ($\lambda_3 = 0.0001$). Hence, from these results we see strong evidence to support that there is specific targeting of sites in Antwerp. On the other hand, the analysis of the London data suggests that there was no targeting of sites in London.

Finally, the counts for the London data are the real counts observed during WWII. The Antwerp data was generated synthetically. Figure 3 (middle) shows the simulated hit locations and square plot counts for the two cities. For the London data, the distribution of hits is uniform. For Antwerp, there are three targeted locations. We can assign each square to one of the components of the mixture, which clusters the squares according to the hit frequency. Figure 3 (bottom) shows the clusters. For the London data, the squares are always assigned to the same component, i.e., there is one dominant component in each mixture, as seen in Figure 3 (top-left). On the other hand, for the Antwerp data, there is a clear distinction between regions. For $K = 2$, the city is separated into regions with 2 or more hits, and those with less than 2. For $K = 3$, the city is separated into regions with 2 or more hits, only 1 hit, and zero hits. This can be seen in the mixture plots, where a different component dominates for different values of x.

.....

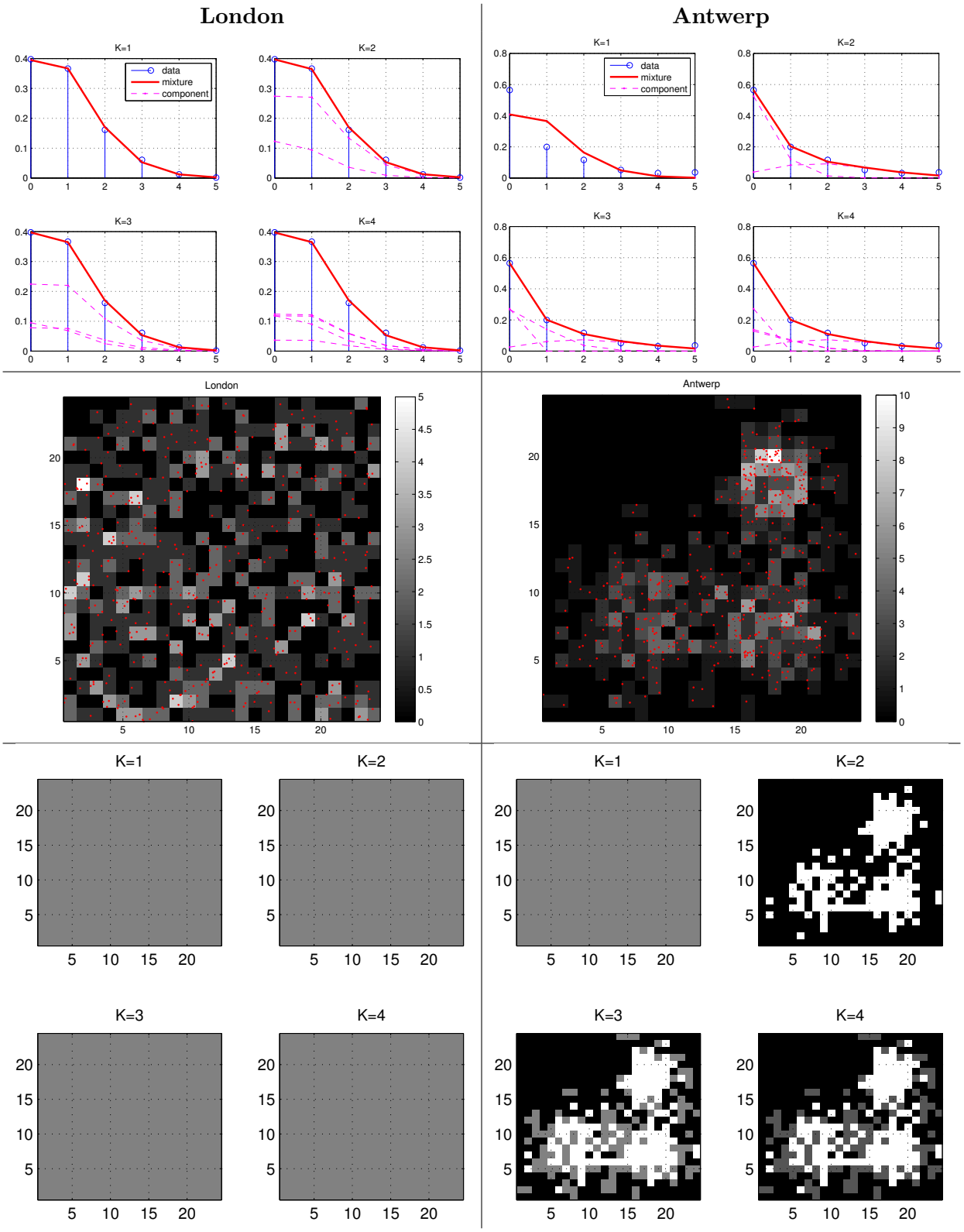

Figure 3: Flying bombs analysis: (top) mixture of Poissons learned for each city, for different number of components (K); (middle) hit locations and square plot counts; (bottom) assignment of squares to most likely components.

Problem 5.2 Mean and variance of a kernel density estimate

(a) The mean of density estimate $\hat{p}(x)$ is

$$\hat{\mu} = \mathbb{E}_{\hat{p}}[x] = \int \hat{p}(x) x dx = \int \frac{1}{n} \sum_{i=1}^{n} \tilde{k}(x - x_i) x dx = \frac{1}{n} \sum_{i=1}^{n} \int \tilde{k}(x - x_i) x dx. \tag{S.69}$$

Looking at the integral, define $\bar{x} = x - x_i$ and perform a change of variable,

$$\int \tilde{k}(x - x_i) x dx = \int \tilde{k}(\bar{x})(\bar{x} + x_i) d\bar{x} = \int \left[\tilde{k}(\bar{x})\bar{x} + \tilde{k}(\bar{x}) x_i \right] d\bar{x} = x_i, \tag{S.70}$$

where the last step follows from the assumption of zero mean of the kernel in (5.6), and the fact that the kernel integrates to 1. Finally, substituting into (S.69),

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{S.71}$$

(b) For the covariance of the density estimate,

$$\hat{\Sigma} = \mathrm{cov}_{\hat{p}}(x) = \mathbb{E}_{\hat{p}} \left[(x - \hat{\mu})(x - \hat{\mu})^T \right] = \int \hat{p}(x) \left[(x - \hat{\mu})(x - \hat{\mu})^T \right] dx \tag{S.72}$$

$$= \int \frac{1}{n} \sum_{i=1}^{n} \tilde{k}(x - x_i) \left[(x - \hat{\mu})(x - \hat{\mu})^T \right] dx = \frac{1}{n} \sum_{i=1}^{n} \int \tilde{k}(x - x_i) \left[(x - \hat{\mu})(x - \hat{\mu})^T \right] dx. \tag{S.73}$$

Looking at the integral, and again performing a change of variable,

$$\int \tilde{k}(x - x_i) \left[(x - \hat{\mu})(x - \hat{\mu})^T \right] dx = \int \tilde{k}(\bar{x}) \left[(\bar{x} + x_i - \hat{\mu})(\bar{x} + x_i - \hat{\mu})^T \right] d\bar{x} \tag{S.74}$$

$$= \int \left[\tilde{k}(\bar{x})\bar{x}\bar{x}^T + \tilde{k}(\bar{x})\bar{x}(x_i - \hat{\mu})^T + \tilde{k}(\bar{x})(x_i - \hat{\mu})\bar{x}^T + \tilde{k}(\bar{x})(x_i - \hat{\mu})(x_i - \hat{\mu})^T \right] d\bar{x} \tag{S.75}$$

$$= H + (x_i - \hat{\mu})(x_i - \hat{\mu})^T, \tag{S.76}$$

where the last step follows from the zero-mean assumption of the kernel, and the fact that the kernel integrates to 1. Finally, substituting into (S.73),

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} \left[H + (x_i - \hat{\mu})(x_i - \hat{\mu})^T \right] = H + \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})(x_i - \hat{\mu})^T. \tag{S.77}$$

(c) The mean of the distribution $\hat{\mu}$ is the same as the empirical mean of the samples. On the other hand, the covariance of the distribution $\hat{\Sigma}$ is the sum of the empirical covariance and the kernel covariance H. Hence, the kernel density estimate produces a distribution with larger covariance than that of the samples. In class, we showed that the mean of the *estimator* is the convolution between the true density and the kernel, $\mathbb{E}_X[\hat{p}(x)] = p(x) * \tilde{k}(x)$, and hence the estimator is biased. From the result in this problem, we can see that this estimator bias manifests itself as a density estimate with covariance larger than what we see empirically in the samples.

.....

Problem 6.4 Coin Tossing

(a) For this problem, the Bayesian decision rule is to guess *heads* (H) when

$$p(s = H|r = H) > p(s = T|r = H) \tag{S.78}$$
$$p(r = H|s = H)p(s = H) > p(r = H|s = T)p(s = T) \tag{S.79}$$
$$(1 - \theta_1)\alpha > \theta_2(1 - \alpha) \tag{S.80}$$
$$\alpha > \frac{\theta_2}{1 - \theta_1 + \theta_2} \tag{S.81}$$

and *tails* (T) when

$$\alpha < \frac{\theta_2}{1 - \theta_1 + \theta_2}. \tag{S.82}$$

When

$$\alpha = \frac{\theta_2}{1 - \theta_1 + \theta_2} \tag{S.83}$$

any guess is equally good.

(b) When $\theta_1 = \theta_2 = \theta$ the minimum probability of error decision is to declare *heads* if

$$\alpha > \theta \tag{S.84}$$

and *tails* otherwise. This means that you should only believe your friend's report if your prior for *heads* is greater than the probability that he lies. To see that this makes sense let's look at a few different scenarios:

- If your friend is a pathological liar ($\theta = 1$), then you know for sure that the answer is not *heads* and you should always say *tails*. This is the decision that (S.84) advises you to take.
- If he never lies ($\theta = 0$) you know that the answer is *heads*. Once again this is the decision that (S.84) advises you to take.
- If both $\alpha = 0$ and $\theta = 0$ we have a contradiction, i.e. you know for sure that the result of the toss is always *tails* but this person that never lies is telling you that it is *heads*. In this case Bayes just gives up and says "either way is fine". This is a sensible strategy, there is something wrong with the models, you probably need to learn something more about the problem.
- If your friend is completely random, $\theta = 1/2$, (S.84) tells you to go with your prior and ignore him. If you believe that that the coin is more likely to land on *heads* say *heads* otherwise say *tails*. Bayes has no problem with ignoring the observations, whenever these are completely uninformative.
- When you do not have prior reason to believe that one of the outcomes is more likely than the other, i.e. if you assume a fair coin ($\alpha = 1/2$), (S.84) advises you to reject the report whenever you think that your friend is more of a liar ($\theta > 1/2$) and to accept it when you believe that he is more on the honest side ($\theta < 1/2$). Once again this makes sense.
- In general, the optimal decision rule is to "modulate" this decision by your prior belief on the outcome of the toss: say *heads* if your prior belief that the outcome was really *heads* is larger than the probability that your friend is lying.

(c) Denoting by r_i the ith report and assuming that the sequence of reports $R = \{r_1, \ldots, r_n\}$ has n_h heads and $n - n_h$ tails, the MPE decision is now to say *heads* if

$$p(s = H|r_1, \ldots, r_n) > p(s = T|r_1, \ldots, r_n) \tag{S.85}$$

$$p(r_1, \ldots, r_n|s = H)p(s = H) > p(r_1, \ldots, r_n|s = T)p(s = T) \tag{S.86}$$

$$(1 - \theta_1)^{n_h}\theta_1^{n-n_h}\alpha > \theta_2^{n_h}(1 - \theta_2)^{n-n_h}(1 - \alpha) \tag{S.87}$$

$$\alpha > \frac{\theta_2^{n_h}(1 - \theta_2)^{n-n_h}}{(1 - \theta_1)^{n_h}\theta_1^{n-n_h} + \theta_2^{n_h}(1 - \theta_2)^{n-n_h}} \tag{S.88}$$

$$\alpha > \frac{1}{1 + (\frac{1-\theta_1}{\theta_2})^{n_h}(\frac{\theta_1}{1-\theta_2})^{n-n_h}} \tag{S.89}$$

$$\tag{S.90}$$

and *tails* otherwise.

(d) When $\theta_1 = \theta_2 = \theta$ and the report sequence is all *heads* ($n_h = n$), the MPE decision becomes to declare *heads* if

$$\alpha > \frac{1}{1 + \left(\frac{1-\theta}{\theta}\right)^n} \tag{S.91}$$

and *tails* otherwise. As n becomes larger, i.e. $n \to \infty$, we have three situations:

- Your friend is more of a liar, $\theta > 1/2$. In this case, $((1 - \theta)/\theta)^n \to 0$ and the decision rule becomes $\alpha > 1$. That is, you should always reject his report.
- Your friend is more of a honest person, $\theta < 1/2$. In this case, $((1 - \theta)/\theta)^n \to \infty$ and the decision rule becomes $\alpha > 0$. That is, you should always accept his report.
- Your friend is really just random, $\theta = 1/2$. In this case, the decision rule becomes $\alpha > 1/2$ and you should go with your prior.

Once again this makes sense. Now you have a lot of observations so you are much more confident on the data and need to rely a lot less on your prior. It also takes a lot less work to figure out what you should do, since you do not have to make detailed probability comparisons. Because your friend seems to be so certain of the outcome (he always says *heads*), you either: 1) not trust him (θ somewhere in between 1/2 and 1) therefore believe that he is just trying to fool you and reject what he says, or 2) trust him (θ somewhere in between 0 and 1/2) and accept his report. It is only in the case that he is completely unpredictable that the prior becomes important. This looks like a really good strategy, and sounds a lot like the way people think. As you can see in this example, the optimal Bayesian decision can be something as qualitative as: *if you trust accept, if you doubt reject, otherwise ignore*.

.....

Problem 7.5 PCA and Classification

(a) To solve this problem in the most general form, we consider that $p(y = 1) = \pi$ and $p(y = 2) = 1 - \pi$. Marginalizing out y, the distribution of the features x is

$$p(x) = \sum_y p(x|y)p(y) = p(x|y = 1)p(y = 1) + p(x|y = 2)p(y = 2) \tag{S.92}$$

$$= \pi p(x|y = 1) + (1 - \pi)p(x|y = 2). \tag{S.93}$$

We then note that, for any function $f(x)$,

$$\mathbb{E}[f(x)] = \int f(x)\pi p(x|y=1)dx + \int f(x)(1-\pi)p(x|y=2)dx \tag{S.94}$$

$$= \pi\mathbb{E}_{x|y=1}[f(x)] + (1-\pi)\mathbb{E}_{x|y=2}[f(x)] \tag{S.95}$$

To compute the mean of x it suffices to apply this result with $f(x) = x$, which leads to

$$\mu_x = \pi\mu_1 + (1-\pi)\mu_2. \tag{S.96}$$

For the covariance we use $f(x) = (x-\mu_x)(x-\mu_x)^T$ leading to

$$\Sigma_x = E[(x-\mu_x)(x-\mu_x)^T] \tag{S.97}$$

$$= \pi\mathbb{E}_{x|y=1}[(x-\mu_x)(x-\mu_x)^T] + (1-\pi)\mathbb{E}_{x|y=2}[(x-\mu_x)(x-\mu_x)^T] \tag{S.98}$$

We next note that

$$\mathbb{E}_{x|y=j}[(x-\mu_x)(x-\mu_x)^T] = \mathbb{E}_{x|y=j}[(x-\mu_j+\mu_j-\mu_x)(x-\mu_j+\mu_j-\mu_x)^T] \tag{S.99}$$

$$= \Sigma_i + \mathbb{E}_{x|y=j}[(x-\mu_j)(\mu_j-\mu_x)^T] + \mathbb{E}_{x|y=j}[(\mu_j-\mu_x)(x-\mu_j)^T] + \mathbb{E}_{x|y=j}[(\mu_j-\mu_x)(\mu_j-\mu_x)^T] \tag{S.100}$$

$$= \Sigma_i + (\mu_i-\mu_x)(\mu_i-\mu_x)^T \tag{S.101}$$

from which it follows that

$$\Sigma_x = \pi\Sigma_1 + (1-\pi)\Sigma_2 + \pi(\mu_1-\mu_x)(\mu_1-\mu_x)^T + (1-\pi)(\mu_2-\mu_x)(\mu_2-\mu_x)^T \tag{S.102}$$

$$= \pi\boldsymbol{\Sigma}_1 + (1-\pi)\boldsymbol{\Sigma}_2 + \pi(1-\pi)^2(\mu_1-\mu_2)(\mu_1-\mu_2)^T + (1-\pi)\pi^2(\mu_1-\mu_2)(\mu_1-\mu_2)^T \tag{S.103}$$

$$= \pi\boldsymbol{\Sigma}_1 + (1-\pi)\boldsymbol{\Sigma}_2 + [\pi(1-\pi)^2 + (1-\pi)\pi^2](\mu_1-\mu_2)(\mu_1-\mu_2)^T \tag{S.104}$$

$$= \pi\boldsymbol{\Sigma}_1 + (1-\pi)\boldsymbol{\Sigma}_2 + \pi(1-\pi)(\mu_1-\mu_2)(\mu_1-\mu_2)^T \tag{S.105}$$

Hence, when $\pi = 1/2$,

$$\mu_x = \mathbb{E}[x] = \frac{1}{2}(\mu_1+\mu_2) \tag{S.106}$$

$$\Sigma_x = \mathbb{E}[(x-\mu_x)(x-\mu_x)^T] = \frac{1}{2}(\Sigma_1+\Sigma_2) + \frac{1}{4}(\mu_1-\mu_2)(\mu_1-\mu_2)^T. \tag{S.107}$$

(b) Regarding PCA, we start by computing the mean and covariance of x. Using the results of (a), we have $\mu_x = 0$ and

$$\Sigma_x = \begin{bmatrix} 1+\alpha^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}. \tag{S.108}$$

Since this is a diagonal matrix, its eigenvectors are just the coordinate axis. Thus PCA will choose the eigenvector $e_1 = [1,0]^T$ when $1+\alpha^2 > \sigma^2$ and the eigenvector $e_2 = [0,1]^T$ otherwise. We thus have two possible decision rules:

- When $\alpha > \sqrt{\sigma^2-1}$ we have $z = e_1^T x = x_1$, where x_1 is the 1st dimension of x and

$$p(z|y=1) = p(x_1 = z|y=1) = \mathcal{N}(z|\alpha, 1) \tag{S.109}$$

$$p(z|y=2) = p(x_1 = z|y=2) = \mathcal{N}(z|-\alpha, 1). \tag{S.110}$$

  This is the direction along which the Gaussians have greatest separation. Hence the classifier obtained with PCA is optimal.

- When $\alpha < \sqrt{\sigma^2 - 1}$ we have $z = e_2^T x = x_2$, and

$$p(z|y=1) = p(x_2 = z|y=1) = \mathcal{N}(z|0, \sigma^2) \tag{S.111}$$

$$p(z|y=2) = p(x_2 = z|y=2) = \mathcal{N}(z|0, \sigma^2) \tag{S.112}$$

  This is a terrible selection of direction to project on, since both classes have the same marginal along this projection and the classification error is therefore going to be the worst possible (0.5, i.e. the same as random guessing). In this case PCA fails miserably.

Overall, the conclusion is that one has to be very careful with the use of PCA for classification. There are situations in which it is optimal, but others in which it leads to the worst possible choice of features. Furthermore, a slight perturbation of the covariance matrix can lead to the switch from one situation to the other. Hence, in the context of classification problems, PCA is not necessarily an optimal technique for dimensionality reduction.

.....

Problem 8.5 Loss functions

(a) Defining $z_i = y_i w^T x_i$ with classes $y_i \in \{+1, -1\}$, the point x_i is misclassified if $z_i < 0$. Hence the number of misclassified training points is the number of points with $z_i < 0$,

$$R_{emp} = \sum_{i=1} L_{01}(z_i), \tag{S.113}$$

where

$$L_{01}(z_i) = \begin{cases} 0, & z_i \geq 0 \\ 1, & z_i < 0 \end{cases}. \tag{S.114}$$

(b) The perceptron minimizes the error over the misclassified points,

$$E(w) = \sum_{i \in \mathcal{M}} -y_i w^T x_i, \tag{S.115}$$

where $\mathcal{M}$ is the set of misclassified points. Noting that correctly classified points (when $z_i \geq 0$) have 0 penalty, the sum can be rewritten over all points as

$$E(w) = \sum_{i \in \mathcal{M}} -z_i = \sum_{i=1}^{n} \begin{cases} 0, & z_i \geq 0 \\ -z_i, & z_i < 0 \end{cases} \tag{S.116}$$

$$= \sum_{i=1}^{n} \max(0, -z_i) = \sum_{i=1}^{n} L_p(z_i), \tag{S.117}$$

where $L_p(z_i) = \max(0, -z_i)$.

(c) Least-squares classification minimizes the squared-error between the label and the linear function,

$$E(w) = \sum_{i=1}^{n} (w^T x_i - y_i)^2 = \sum_{i=1}^{n} \left(\frac{1}{y_i} (y_i w^T x_i - y_i^2) \right)^2 = \sum_{i=1}^{n} \frac{1}{y_i^2} \left(y_i w^T x_i - y_i^2 \right)^2 \tag{S.118}$$

$$= \sum_{i=1}^{n} (y_i w^T x_i - 1)^2 = \sum_{i=1}^{n} (z_i - 1)^2, \tag{S.119}$$

where the last line follows from $y_i^2 = 1$. Hence, $L_{LSC}(z_i) = (z_i - 1)^2$.

(d) Logistic regression maximizes the data log-likelihood, or equivalently minimizes the "cross-entropy" error

$$E(w) = -\sum_{i=1}^{n} [\tilde{y}_i \log \pi_i + (1 - \tilde{y}_i) \log(1 - \pi_i)] \tag{S.120}$$

where the classes are now defined as $\tilde{y}_i \in \{1, 0\}$, and $\pi_i = \sigma(w^T x_i)$ is the classifier's probability (confidence) that x_i should be in class $\tilde{y}_i = 1$. Because $\tilde{y}_i$ can only take a value of either 1 or 0, only one of the terms in the square brackets is non-zero for each i. Hence, we have

$$E(w) = -\sum_{i=1}^{n} \begin{cases} \log \pi_i, & \tilde{y}_i = 1 \\ \log(1 - \pi_i), & \tilde{y}_i = 0 \end{cases} \tag{S.121}$$

Next, we map the class labels from $\tilde{y}_i \in \{1, 0\}$ to $y_i \in \{+1, -1\}$, and the error function is

$$E(w) = -\sum_{i=1}^{n} \begin{cases} \log \pi_i, & y_i = +1 \\ \log(1 - \pi_i), & y_i = -1 \end{cases} \tag{S.122}$$

$$= -\sum_{i=1}^{n} \begin{cases} \log \sigma(w^T x_i), & y_i = +1 \\ \log \sigma(-w^T x_i), & y_i = -1 \end{cases} \tag{S.123}$$

$$= -\sum_{i=1}^{n} \log \sigma(y_i w^T x_i) \tag{S.124}$$

$$= \sum_{i=1}^{n} -\log \frac{1}{1 + \exp(-y_i w^T x_i)} = \sum_{i=1}^{n} \log(1 + e^{-z_i}), \tag{S.125}$$

where (S.123) follows from the property $1 - \sigma(f) = \sigma(-f)$ from Problem 8.1. Hence, the logistic regression loss function is

$$L_{LR}(z_i) = \log(1 + e^{-z_i}) \propto \frac{1}{\log(2)} \log(1 + e^{-z_i}). \tag{S.126}$$

The purpose of the scaling term $1/\log(2)$ is to make the loss function go through the (0,1) "corner" of the 0-1 loss function; obviously it doesn't change the underlying optimization problem.

(e) See discussion for Problem 9.5 (c).

. . . . . .

Problem 9.1 Margin

(a) The goal is to find the point x on the hyperplane $f(x) = w^T x + b = 0$ that is closest to a point x_a. In other words, we want to minimize the distance from x to x_a, subject to the constraint $f(x) = 0$. The Lagrangian for this problem is

$$L = \|x - x_a\|^2 - \lambda(w^T x + b). \tag{S.127}$$

Setting the derivative wrt λ to 0, we recover the desired equality constraint

$$\frac{\partial L}{\partial \lambda} = -(w^T x + b) = 0 \quad \Rightarrow \quad f(x) = 0. \tag{S.128}$$

Next setting the derivative wrt x to 0,

$$\frac{\partial L}{\partial x} = 2(x - x_a) - \lambda w = 0 \quad \Rightarrow \quad 2(x - x_a) = \lambda w. \tag{S.129}$$

Multiplying both sides by w^T,

$$2w^T(x - x_a) = \lambda w^T w \tag{S.130}$$
$$2(w^T x - w^T x_a) = \lambda \|w\|^2 \tag{S.131}$$
$$2(w^T x + b - w^T x_a - b) = \lambda \|w\|^2 \tag{S.132}$$
$$2(f(x) - f(x_a)) = \lambda \|w\|^2 \tag{S.133}$$
$$\Rightarrow \lambda = \frac{-2f(x_a)}{\|w\|^2}, \tag{S.134}$$

which follows from the constraint $f(x) = 0$. Next substituting for λ in (S.129),

$$(x - x_a) = \frac{-f(x_a)}{\|w\|^2} w. \tag{S.135}$$

The length-squared of this vector is

$$\|x - x_a\|^2 = \frac{f(x_a)^2}{\|w\|^4} \|w\|^2, \tag{S.136}$$

and hence the minimal distance between x (a point on the hyperplane) and x_a is

$$\|x - x_a\| = \frac{|f(x_a)|}{\|w\|}, \tag{S.137}$$

which is the distance from x_a to the hyperplane.

(b) Setting $x_a = 0$ as the origin, we have

$$\|x\| = \frac{|w^T 0 + b|}{\|w\|} = \frac{|b|}{\|w\|}, \tag{S.138}$$

which is the distance from the origin to the hyperplane.

(c) The point closest to x_a on the hyperplane is obtained from (S.135),

$$\hat{x} = x_a - \frac{f(x_a)}{\|w\|^2} w. \tag{S.139}$$

.....

Problem 9.5 Soft-margin SVM risk function

(a) The soft-margin constraint is

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad \Rightarrow \quad \xi_i \geq 1 - y_i(w^T x_i + b). \tag{S.140}$$

The slack variable is also non-negative, $\xi_i \geq 0$. Both inequalities are a lower bound on ξ_i, and hence they can be combined by taking the maximum of the two

$$\xi_i \geq \max(0, 1 - y_i(w^T x_i + b)). \tag{S.141}$$

Note that the soft-margin objective,

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i, \tag{S.142}$$

is minimizing the 1-norm of ξ_i. Hence, at the minimum, we must have equality for the constraint in (S.141) (i.e., the constraint is active),

$$\xi_i = \max(0, 1 - y_i(w^T x_i + b)). \tag{S.143}$$

If this were not true, we could still reduce ξ_i and satisfy the inequality constraint, which means the objective is not minimized yet.

(b) Substituting (S.143) into the objective in (S.142), we have the error function

$$E(w, b) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i(w^T x_i + b)) \tag{S.144}$$

$$\propto \sum_{i=1}^{n} \max(0, 1 - y_i(w^T x_i + b)) + \lambda \|w\|^2, \tag{S.145}$$

where $\lambda = 1/C > 0$. The left term is the empirical risk, where the loss function is

$$L_{SVM}(z_i) = \max(0, 1 - y_i(w^T x_i + n)) = \max(0, 1 - z_i), \tag{S.146}$$

and the right term regularizes w (i.e., controls the complexity). This loss function is also known as the "hinge loss".

(c) Figure 4 plots the loss function for various classifiers: perceptron, least-squares (LS), logistic regression (LR), SVM, and AdaBoost. We didn't talk about boosting (AdaBoost) in this class, but anyways, it can be shown to be minimizing an exponential loss function $L_{exp}(z_i) = e^{-z_i}$. There are three important regions in this plot: misclassified points ($z < 0$), correctly classified points inside the margin $0 < z < 1$, and correct points outside the margin $1 < z$. Let's compare these regions for the different classifiers.

- For points that are misclassified ($z < 0$), the perceptron, LR, and SVM all apply a penalty that is linear to the distance of the point from the classification boundary. On the other hand, boosting and LS both use non-linear penalties for misclassifications, with boosting being the most aggressive (exponential penalty). Intuitively, classifiers with the linear misclassification penalty (LR, SVM) will be more robust to outliers and errors caused by overlapping classes. On the other hand, boosting will tend to overfit to reduce these errors (in the absence of enough training data).

- All the classifiers except for the perceptron penalize correctly classified points that are inside the margin ($0 \leq z \leq 1$). The main purpose is to find a hyperplane that has the most amount of "buffer" space (neutral zone) between the positive and negative classes. The perceptron is very similar to the SVM, except that it doesn't have a margin penalty!
- Finally, the classification methods differ in how they deal with correctly classified points far from the margin ($z > 1$). First, for SVM, there is no penalty if the point falls outside the margin, i.e., these points don't affect the hyperplane at all. On the other hand, LR and boosting apply an exponential decaying penalty, which causes correctly classified points to be "pushed" farther away from the margin (actually the points are stationary and the margin is being pushed). This has the effect of tilting the decision boundary away from the bulk of the data (see Figure 5 for an example). Finally, LS applies a large penalty for points that are correctly classified and far from the margin, i.e. that are "too correct". So, in actuality, LS is pushing all the points towards the margin, regardless if they are correct or not.

So which classifiers are better? Well, there are different properties that might be desirable for a classifier: 1) robustness to outliers (linear misclassification penalty); 2) maximizing the margin; 2) and pushing points away from the margin. These properties are embodied in the three classifiers: SVM, LR, and AdaBoost. Interestingly, LR is somehow a mix of SVM (with linear misclassification penalty) and AdaBoost (with exponential decaying penalty for correct classifications).

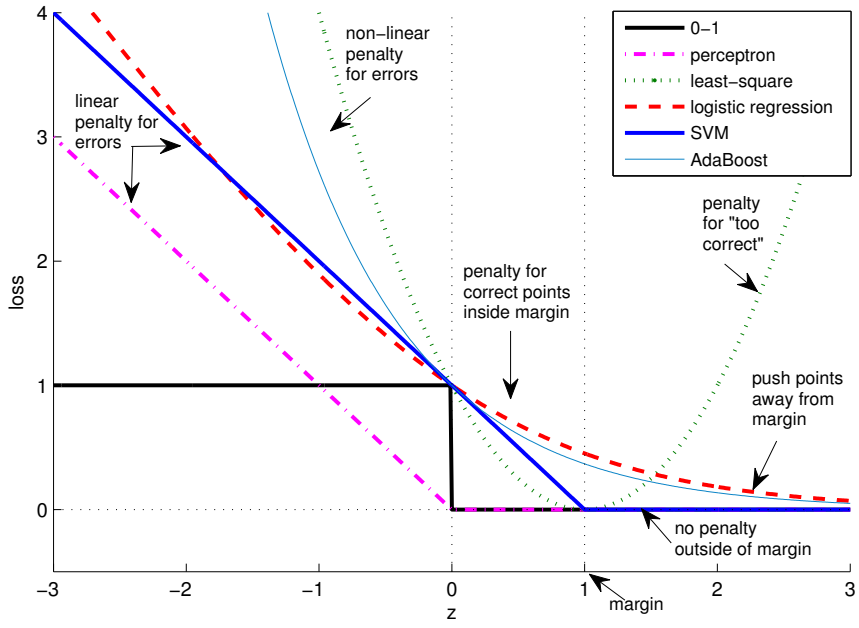

Figure 4: Loss functions versus distance to margin ($z = yw^T x$)

.....

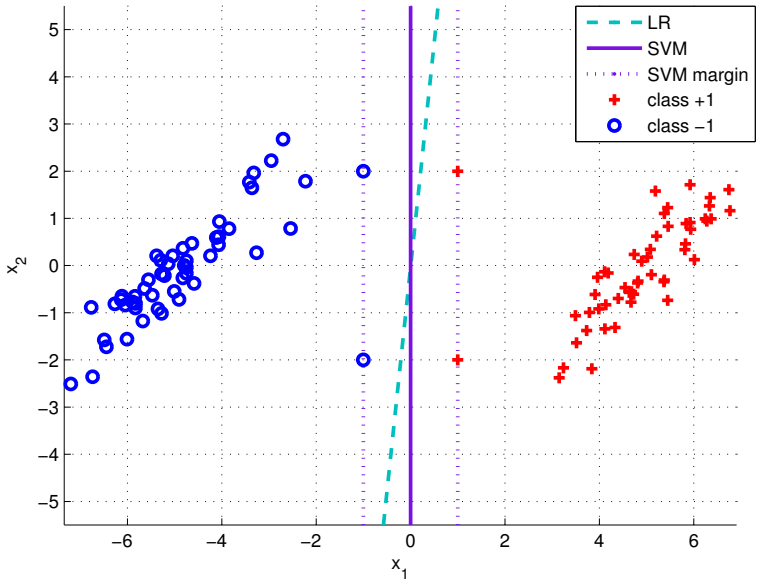

Figure 5: Example of logistic regression and SVM classification boundaries. The LR boundary is slightly tilted away from the data, whereas the SVM boundary only depends on the points on the margin.

Problem 10.1 Constructing kernels from kernels

(f) The Taylor expansion (at 0) of the exponential function is

$$e^a = \sum_{n=0}^{\infty} \frac{a^n}{n!} = 1 + a + \frac{a^2}{2} + \frac{a^3}{6} + \frac{a^4}{24} + \cdots \tag{S.147}$$

Hence, substituting the kernel function for a

$$k(x, z) = e^{k_1(x,z)} = \sum_{n=0}^{\infty} \frac{1}{n!} k_1(x, z)^n. \tag{S.148}$$

This is an infinite sum of non-negative integer powers of the kernel function, where each term is scaled by a positive value. If $k_1(x, z)$ is a positive definite kernel, then all these operations also result in a positive definite kernel (see Problems 10.1a, 10.1b, and 10.1e). Hence, the exponential of a positive definite kernel is also positive definite.

.....

Problem 10.2 Kernels on real vectors

(a) We can rewrite the RBF as

$$k(x, z) = e^{-\alpha\|x-z\|^2} = e^{-\alpha(x^T x - 2x^T z + z^T z)} = e^{-\alpha x^T x} e^{2\alpha x^T z} e^{-\alpha z^T z} \tag{S.149}$$

The middle term $k_1(x, z) = e^{2\alpha x^T z}$ is a positive definite kernel via Problem 10.1(f) and the fact that the linear kernel $x^T z$ is positive definite (as is the scaled version). Next letting

$f(x) = e^{-\alpha x^T x}$, we can rewrite the RBF kernel as

$$k(x, z) = f(x)e^{2\alpha x^T z} f(z). \tag{S.150}$$

Hence, using the input scaling property (Problem 10.1(d)), the RBF kernel is also positive definite.

In lecture we saw that the mapping for the RBF kernel is from a point to a function, $x_i \to k(\cdot, x_i) = e^{-\alpha\|\cdot - x_i\|^2}$, in the RKHS. Using the Taylor expansion of the exponential function we can get an alternative interpretation. For simplicity, consider the case when $\alpha = 1/2$. The exponential term can be written as a sum of polynomial terms,

$$k_1(x, z) = e^{x^T z} = \sum_{n=0}^{\infty} \frac{1}{n!}(x^T z)^n = 1 + x^T z + \frac{1}{2}(x^T z)^2 + \frac{1}{6}(x^T z)^3 + \frac{1}{24}(x^T z)^4 + \cdots \tag{S.151}$$

Letting $\Phi_p(x)$ be the feature transform of the degree-p polynomial,

$$k_1(x, z) = 1 + x^T z + \frac{1}{2}\Phi_2(x)^T\Phi_2(z) + \frac{1}{6}\Phi_3(x)^T\Phi_3(z) + \frac{1}{24}\Phi_4(x)^T\Phi_4(z) + \cdots \tag{S.152}$$

$$= \Phi(x)^T\Phi(z), \quad \Phi(x) = \begin{bmatrix} 1 \\ x \\ \frac{1}{\sqrt{2}}\Phi_2(x) \\ \frac{1}{\sqrt{6}}\Phi_3(x) \\ \frac{1}{\sqrt{24}}\Phi_4(x) \\ \vdots \end{bmatrix} \tag{S.153}$$

Hence, the feature transformation of the exponential term is a concatenation of the feature transformations of all the polynomial kernels (i.e., a concatenation of all polynomial combination of features) with decaying weights. Note that the input scaling terms are

$$f(x) = e^{-\alpha x^T x} = \left[e^{2\alpha x^T x}\right]^{-1/2} = \frac{1}{\sqrt{k_1(x, x)}}. \tag{S.154}$$

Hence, the RBF kernel is the normalized version of the exponential kernel k_1, which only measures the angle between the high-dimensional vectors in feature space (see Problem 10.4 for more about normalized kernels).

.....

Problem 10.4 Constructing kernels from kernels (part 3 – normalization)

(a) Setting $f(x) = \frac{1}{\sqrt{k(x,x)}}$ and using the "input scaling" property (Problem 10.1(d)) shows that the normalized kernel is positive definite if $k(x, z)$ is also positive definite.

(b) The positive definite kernel $k(x, z)$ can be expressed as an inner-product in the high-dimensional space, $k(x, z) = \langle\Phi(x), \Phi(z)\rangle$. The length (norm) of the transformed vector is $\|\Phi(x)\| = \sqrt{\langle\Phi(x), \Phi(x)\rangle}$. Hence, the normalized kernel can be written as

$$\tilde{k}(x, z) = \frac{k(x, z)}{\sqrt{k(x, x)k(z, z)}} = \frac{\langle\Phi(x), \Phi(z)\rangle}{\|\Phi(x)\| \, \|\Phi(z)\|}, \tag{S.155}$$

which is the formula for the cosine of the angle between $\Phi(x)$ and $\Phi(z)$. Usually this relationship is written as: $\langle x, z\rangle = \|x\| \, \|z\| \cos\theta$.

(c) Since $\tilde{k}(x, z)$ is the cosine of an angle, we have $-1 \leq \tilde{k}(x, z) \leq 1$. In addition $\tilde{k}(x, z) = 1$ when $\Phi(x)$ and $\Phi(z)$ are pointing in the same direction, $\tilde{k}(x, z) = -1$ if they are pointing in opposite directions, and $\tilde{k}(x, z) = 0$ when $\Phi(x)$ is orthogonal to $\Phi(z)$.

On a side note, squaring the normalized kernel, we can obtain the inequality

$$\tilde{k}(x, z)^2 \leq 1 \quad \Rightarrow \quad k(x, z)^2 \leq k(x, x)k(z, z), \tag{S.156}$$

which is true for any positive definite kernel $k(x, z)$. Taking the square-root, we obtain the Cauchy-Schwarz inequality for kernels,

$$|k(x, z)| \leq \sqrt{k(x, x)k(z, z)}. \tag{S.157}$$

Finally, Figure 6 shows the exponential kernel matrix for the data in Figure 5, and its normalized kernel (i.e., RBF kernel). In the unnormalized kernel, some rows have within-class kernel values that are similar to out-of-class values for other rows. The dynamic range of the similarity values changes depending on the location, which makes the learning task harder for the classifier. On the other hand, after normalizing the kernel, the maximum similarity is 1 (on the diagonal), and all similarities in each row are scaled relative to this.

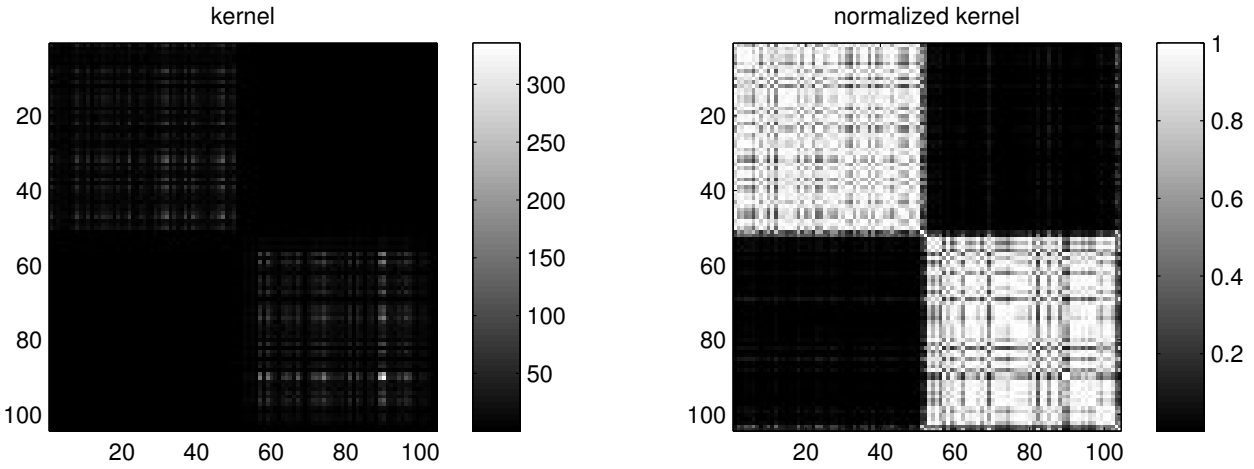

Figure 6: (left) the exponential kernel $k(x, z) = e^{\alpha x^T z}$; (right) the normalized exponential kernel, i.e. RBF kernel. The data is from Figure 5 and $\alpha = 0.1$.

. . . . . .

Problem 1.6 Multivariate Gaussian

(a) The multivariate Gaussian distribution is

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}\,|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right\}. \tag{S.158}$$

Assuming a diagonal covariance matrix, $\Sigma = \begin{bmatrix}\sigma_1^2 & & 0\\ & \ddots & \\ 0 & & \sigma_d^2\end{bmatrix}$, and substituting the properties in (1.16),

$$\mathcal{N}(x|\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}(\prod_{i=1}^d \sigma_i^2)^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T \begin{bmatrix}1/\sigma_1^2 & & 0\\ & \ddots & \\ 0 & & 1/\sigma_d^2\end{bmatrix}(x-\mu)\right\} \tag{S.159}$$

$$= \frac{1}{(2\pi)^{d/2}(\prod_{i=1}^d \sigma_i)} \exp\left\{-\frac{1}{2}\sum_{i=1}^d \frac{1}{\sigma_i^2}(x_i-\mu_i)^2\right\} \tag{S.160}$$

$$= \prod_{i=1}^d \left[\frac{1}{(2\pi)^{1/2}\sigma_i} \exp\left\{-\frac{1}{2}\frac{1}{\sigma_i^2}(x_i-\mu_i)^2\right\}\right] \tag{S.161}$$

$$= \prod_{i=1}^d \mathcal{N}(x_i|\mu_i,\sigma_i^2) \tag{S.162}$$

(b) See Figure 7b. The diagonal terms indicate how far the Gaussian stretches in each axis direction.

(c) See Figure 7a.

(d) The eigenvalues and eigenvector pairs (λ_i, v_i) of Σ satisfy

$$\Sigma v_i = \lambda_i v_i, \quad i \in \{1,\cdots,d\}. \tag{S.163}$$

Rewriting using matrix notation,

$$\Sigma\begin{bmatrix}v_1 & \cdots & v_d\end{bmatrix} = \begin{bmatrix}\lambda_1 v_1 & \cdots & \lambda_d v_d\end{bmatrix} \tag{S.164}$$

$$\Sigma\begin{bmatrix}v_1 & \cdots & v_d\end{bmatrix} = \begin{bmatrix}v_1 & \cdots & v_d\end{bmatrix}\cdot\begin{bmatrix}\lambda_1 & & 0\\ & \ddots & \\ 0 & & \lambda_d\end{bmatrix} \tag{S.165}$$

$$\Sigma V = V\Lambda, \tag{S.166}$$

where $V = \begin{bmatrix}v_1 & \cdots & v_d\end{bmatrix}$ is a matrix of eigenvectors, and $\Lambda = \begin{bmatrix}\lambda_1 & & 0\\ & \ddots & \\ 0 & & \lambda_d\end{bmatrix}$ is a diagonal matrix with the corresponding eigenvalues. Finally, post-multiplying both sides by V^{-1},

$$\Sigma V V^{-1} = V\Lambda V^{-1} \tag{S.167}$$

$$\Sigma = V\Lambda V^T, \tag{S.168}$$

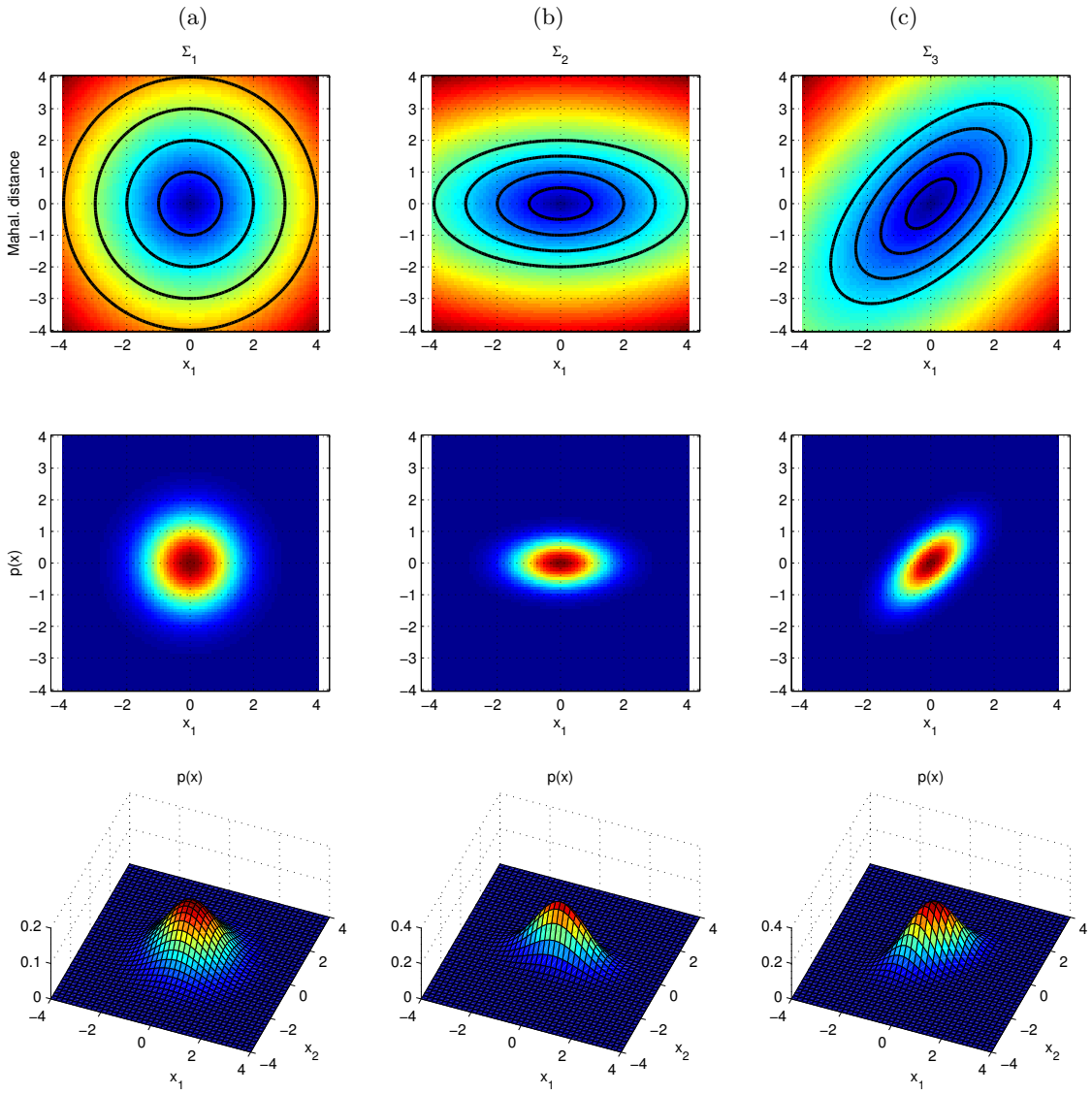

Figure 7: Example of multivariate Gaussians: a) isotropic covariance matrix; b) diagonal covariance matrix; c) full covariance matrix.

where in the last line we have used the property that the eigenvectors are orthogonal, i.e., $V^T V = I$ and hence $V^{-1} = V^T$.

(e) The inverse of the covariance matrix is

$$\Sigma^{-1} = (V\Lambda V^T)^{-1} = V^{-T}\Lambda^{-1}V^{-1} = V\Lambda^{-1}V^T. \tag{S.169}$$

Hence, the Mahalanobis distance term can be rewritten as

$$\|x-\mu\|_\Sigma^2 = (x-\mu)^T\Sigma^{-1}(x-\mu) = (x-\mu)V\Lambda^{-1}\underbrace{V^T(x-\mu)}_{y} = y\Lambda^{-1}y = \|y\|_\Lambda^2\,, \tag{S.170}$$

where we define $y = V^T(x-\mu)$.

(f) In the transformation $x = Vy + \mu$, first y is rotated according to the eigenvector matrix V, then the result is translated by vector μ.

(g) See Figure 7c. For this Σ, we have $V = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{-\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$ and $\Lambda = \begin{bmatrix} 1 & 0 \\ 0 & 0.25 \end{bmatrix}$. The Gaussian is first stretched according to the eigenvalues (e.g., like in Figure 7b). Then it is rotated to match the directions of the eigenvectors.

.....

Problem 1.8 Product of Multivariate Gaussian Distributions

The product of two Gaussians is

$$\mathcal{N}(x|a, A)\mathcal{N}(x|b, B) \tag{S.171}$$

$$= \frac{1}{(2\pi)^{d/2}\,|A|^{1/2}} \exp\left\{-\frac{1}{2}(x-a)^T A^{-1}(x-a)\right\} \frac{1}{(2\pi)^{d/2}\,|B|^{1/2}} \exp\left\{-\frac{1}{2}(x-b)^T B^{-1}(x-b)\right\} \tag{S.172}$$

$$= \frac{1}{(2\pi)^{d}\,|A|^{1/2}\,|B|^{1/2}} \exp\left\{-\frac{1}{2}\left[(x-a)^T A^{-1}(x-a) + (x-b)^T B^{-1}(x-b)\right]\right\}. \tag{S.173}$$

Looking at the exponent term, we expand and collect terms,

$$M = (x-a)^T A^{-1}(x-a) + (x-b)^T B^{-1}(x-b) \tag{S.174}$$

$$= x^T A^{-1} x + x^T B^{-1} x - 2a^T A^{-1} x - 2b^T B^{-1} x + a^T A^{-1} a + b^T B^{-1} b \tag{S.175}$$

$$= x^T \underbrace{(A^{-1} + B^{-1})}_{\mathbf{A}} x - 2\underbrace{(a^T A^{-1} + b^T B^{-1})}_{\mathbf{b^T}} x + \underbrace{a^T A^{-1} a + b^T B^{-1} b}_{\mathbf{c}}. \tag{S.176}$$

Using the above definitions of $(\mathbf{A}, \mathbf{b}, \mathbf{c})$ we complete the square using Problem 1.10:

$$\mathcal{M} = (x - \mathbf{d})^T \mathbf{A}(x - \mathbf{d}) + \mathbf{e}, \tag{S.177}$$

where

$$\mathbf{d} = \mathbf{A}^{-1}\mathbf{b} = (A^{-1} + B^{-1})^{-1}(A^{-1}a + B^{-1}b) \tag{S.178}$$

and

$$\mathbf{e} = \mathbf{c} - \mathbf{b}^T \mathbf{A}^{-1} \mathbf{b} \tag{S.179}$$

$$= a^T A^{-1} a + b^T B^{-1} b - (a^T A^{-1} + b^T B^{-1})(A^{-1} + B^{-1})^{-1}(A^{-1}a + B^{-1}b) \tag{S.180}$$

$$= a^T A^{-1} a - a^T A^{-1}(A^{-1} + B^{-1})^{-1} A^{-1} a + b^T B^{-1} b - b^T B^{-1}(A^{-1} + B^{-1})^{-1} B^{-1} b - 2a^T A^{-1}(A^{-1} + B^{-1})^{-1} B^{-1} b \tag{S.181}$$

$$= a^T (A + B)^{-1} a + b^T (A + B)^{-1} b - 2a^T (A + B)^{-1} b \tag{S.182}$$

$$= (a - b)^T (A + B)^{-1}(a - b), \tag{S.183}$$

where in (S.182) we use the matrix inversion lemma on the first two terms (from Problem 1.15). Finally, defining $C = (A^{-1} + B^{-1})^{-1}$ and $c = C(A^{-1}a + B^{-1}b)$, we obtain for the exponent term

$$\mathcal{M} = (x-c)^T C^{-1}(x-c) + (a-b)^T (A+B)^{-1}(a-b) = \|x-c\|_C^2 + \|a-b\|_{A+B}^2 . \tag{S.184}$$

Next, we look at the determinant term,

$$\mathcal{D} = \frac{1}{|A|^{1/2}\,|B|^{1/2}} = \frac{1}{|A|^{1/2}\,|B|^{1/2}} \frac{|C|^{1/2}}{|C|^{1/2}} \tag{S.185}$$

$$= \frac{1}{|A|^{1/2}\,|B|^{1/2}} \frac{\left|(A^{-1}+B^{-1})^{-1}\right|^{1/2}}{|C|^{1/2}} \tag{S.186}$$

$$= \frac{1}{|A|^{1/2}\,|B|^{1/2}\,|A^{-1}+B^{-1}|^{1/2}} \frac{1}{|C|^{1/2}} \tag{S.187}$$

$$= \frac{1}{(|A|\,|A^{-1}+B^{-1}|\,|B|)^{1/2}} \frac{1}{|C|^{1/2}} \tag{S.188}$$

$$= \frac{1}{|A+B|^{1/2}} \frac{1}{|C|^{1/2}} \tag{S.189}$$

Finally, substituting the derived expressions for $\mathcal{M}$ and $\mathcal{D}$ into (S.173)

$$\mathcal{N}(x|a,A)\mathcal{N}(x|b,B) \tag{S.190}$$

$$= \frac{1}{(2\pi)^d\,|A+B|^{1/2}} \frac{1}{|C|^{1/2}} \exp\left\{-\frac{1}{2}\left[\|x-c\|_C^2 + \|a-b\|_{A+B}^2\right]\right\} \tag{S.191}$$

$$= \left[\frac{1}{(2\pi)^{d/2}\,|A+B|^{1/2}} \exp\left\{-\frac{1}{2}\|a-b\|_{A+B}^2\right\}\right] \left[\frac{1}{(2\pi)^{d/2}\,|C|^{1/2}} \exp\left\{-\frac{1}{2}\|x-c\|_C^2\right\}\right] \tag{S.192}$$

$$= \mathcal{N}(a|b,A+B)\mathcal{N}(x|c,C) \tag{S.193}$$

.....

Problem 1.10 Completing the square

The original form is

$$f(x) = x^T A x - 2x^T b + c. \tag{S.194}$$

Now expand the desired form,

$$f(x) = (x-d)^T A(x-d) + e = \underbrace{x^T A x}_{\text{quadratic}} - \underbrace{2x^T A d}_{\text{linear}} + \underbrace{d^T A d + e}_{\text{constant}} . \tag{S.195}$$

We need to match the quadratic, linear, and constant terms. The quadratic term already matches. For the linear term, we have by inspection $d = A^{-1}b$ so that $x^T A d = x^T b$. For the constant term, we set

$$c = d^T A d + e \quad \Rightarrow \quad e = c - d^T A d \tag{S.196}$$

$$= c - b^T A^{-1} A A^{-1} b = c - b^T A^{-1} b \tag{S.197}$$

.....

Problem 2.6 MLE for a multivariate Gaussian

(a) The log-likelihood of the data is

$$\ell = \log p(X) = \sum_{i=1}^{N} \log p(x_i) = \sum_{i=1}^{N} \left\{ -\frac{1}{2} \|x_i - \mu\|_\Sigma^2 - \frac{1}{2} \log |\Sigma| - \frac{d}{2} \log 2\pi \right\} \tag{S.198}$$

$$= -\frac{1}{2} \sum_{i=1}^{N} \|x_i - \mu\|_\Sigma^2 - \frac{N}{2} \log |\Sigma| \tag{S.199}$$

$$= -\frac{1}{2} \sum_{i=1}^{N} (x_i - \mu)^T \Sigma^{-1} (x_i - \mu) - \frac{N}{2} \log |\Sigma| \,, \tag{S.200}$$

where we have dropped constant terms. To find the maximum of the log-likelihood ℓ w.r.t. μ, we take the derivative and set to 0. One way to proceed is to expand the quadratic term and take the derivative. Alternatively, we can use the chain rule. First, removing terms that do not depend on μ,

$$\frac{\partial \ell}{\partial \mu} = -\frac{1}{2} \sum_{i=1}^{N} \frac{\partial}{\partial \mu} (x_i - \mu)^T \Sigma^{-1} (x_i - \mu). \tag{S.201}$$

Letting $z_i = x_i - \mu$ and applying the chain rule $\frac{\partial f}{\partial \mu} = \frac{\partial f}{\partial z} \frac{\partial z}{\partial \mu}$,

$$\frac{\partial \ell}{\partial \mu} = -\frac{1}{2} \sum_{i=1}^{N} \frac{\partial}{\partial \mu} z_i^T \Sigma^{-1} z_i = -\frac{1}{2} \sum_{i=1}^{N} \left[\frac{\partial}{\partial z_i} z_i^T \Sigma^{-1} z_i \right] \left[\frac{\partial z_i}{\partial \mu} \right] \tag{S.202}$$

$$= -\frac{1}{2} \sum_{i=1}^{N} \left[2\Sigma^{-1} z_i \right] [-1] = \sum_{i=1}^{N} \Sigma^{-1} (x_i - \mu). \tag{S.203}$$

Setting the derivative to 0, and solving for μ,

$$\sum_{i=1}^{N} \Sigma^{-1} (x_i - \mu) = 0. \tag{S.204}$$

Pre-multiplying both sides by Σ,

$$\Sigma \sum_{i=1}^{N} \Sigma^{-1} (x_i - \mu) = \Sigma 0 \quad \Rightarrow \quad \sum_{i=1}^{N} x_i - \sum_{i=1}^{N} \mu = 0 \tag{S.205}$$

$$\Rightarrow \quad \sum_{i=1}^{N} x_i - N\mu = 0 \quad \Rightarrow \quad \hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{S.206}$$

(b) To find the maximum of the log-likelihood w.r.t. Σ, we first use the "trace" trick, $x^T A x =$

$\text{tr}[x^T Ax] = \text{tr}[Axx^T]$, on the log-likelihood,

$$\ell = -\frac{1}{2}\sum_{i=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu) - \frac{N}{2}\log|\Sigma| \tag{S.207}$$

$$= -\frac{1}{2}\sum_{i=1}^{N}\text{tr}[\Sigma^{-1}(x_i-\mu)(x_i-\mu)^T] - \frac{N}{2}\log|\Sigma|\,, \tag{S.208}$$

$$= -\frac{1}{2}\text{tr}\left[\Sigma^{-1}\sum_{i=1}^{N}(x_i-\mu)(x_i-\mu)^T\right] - \frac{N}{2}\log|\Sigma|\,. \tag{S.209}$$

Taking the derivative w.r.t. Σ (using the helpful derivatives provided in Problem 2.6) and setting to 0,

$$\frac{\partial\ell}{\partial\Sigma} = -\frac{1}{2}\left\{-\Sigma^{-1}\left[\sum_{i=1}^{N}(x_i-\mu)(x_i-\mu)^T\right]\Sigma^{-1}\right\} - \frac{N}{2}\Sigma^{-1} = 0. \tag{S.210}$$

Pre-multiplying and post-multiplying by Σ on both sides gives

$$\frac{1}{2}\left[\sum_{i=1}^{N}(x_i-\mu)(x_i-\mu)^T\right] - \frac{N}{2}\Sigma = 0 \tag{S.211}$$

$$\Rightarrow \quad \hat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)(x_i-\mu)^T. \tag{S.212}$$

.....

Problem 2.8 Least-squares regression and MLE

(a) We want to find the θ that minimizes the sum-squared error,

$$E = \sum_{i=1}^{N}(y_i-\phi(x_i))^2 = \left\|y-\Phi^T\theta\right\|^2 = (y-\Phi^T\theta)^T(y-\Phi^T\theta) \tag{S.213}$$

$$= y^Ty - 2y^T\Phi^T\theta + \theta^T\Phi\Phi^T\theta. \tag{S.214}$$

Next, take the derivative w.r.t. θ (using the derivatives from Problem 2.6), and setting to zero,

$$\frac{\partial E}{\partial\theta} = -2\Phi y + 2\Phi\Phi^T\theta = 0 \tag{S.215}$$

$$\Rightarrow \quad \Phi\Phi^T\theta = \Phi y \quad \Rightarrow \quad \hat{\theta} = (\Phi\Phi^T)^{-1}\Phi y. \tag{S.216}$$

(b) The noise ϵ is zero-mean Gaussian, and hence $y_i = f(x_i) + \epsilon$ is also Gaussian with mean equal to the function value $f(x_i) = \phi(x_i)^T\theta$,

$$p(y_i|x_i,\theta) = \mathcal{N}(y_i|\phi(x_i)^T\theta, \sigma^2). \tag{S.217}$$

Then the log-likelihood of the data is

$$\ell = \sum_{i=1}^{N}\log p(y_i|x_i\theta) = \sum_{i=1}^{N}\left\{-\frac{1}{2\sigma^2}(y_i-\phi(x_i)^T\theta)^2 - \frac{1}{2}\log\sigma^2 - \frac{1}{2}\log 2\pi\right\}. \tag{S.218}$$

The maximum likelihood solution is then

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \ell = \underset{\theta}{\operatorname{argmax}} -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \phi(x_i)^T \theta)^2 = \operatorname{argmin} \sum_{i=1}^{N} (y_i - \phi(x_i)^T \theta)^2, \tag{S.219}$$

where the last step follows from $\frac{-1}{2\sigma^2}$ being a scalar constant w.r.t. θ. Hence, the maximum likelihood solution is equivalent to the least-squares solution. This is mainly because we assumed Gaussian noise, which introduces the squared-error term.

.....

Problem 3.10 Bayesian regression with Gaussian prior

(a) The posterior distribution of the parameters is obtained using Bayes' rule,

$$p(\theta|y, X) = \frac{p(y, |X, \theta)p(\theta)}{\int p(y|X, \theta)p(\theta)d\theta}. \tag{S.220}$$

Here the denominator ensures that the posterior is properly normalized (integrates to 1). Note that the denominator is only a function of the data $\mathcal{D}$ since the parameter θ is integrated out. Hence, it suffices to find the form of θ in the numerator first, and then normalize that equation to obtain the distribution,

$$p(\theta|y, X) \propto p(y|X, \theta)p(\theta) \tag{S.221}$$

or equivalently

$$\log p(\theta|y, X) = \log p(y|X, \theta) + \log p(\theta) + \text{const.} \tag{S.222}$$

Substituting for the data likelihood and prior terms (and ignoring terms not involving θ),

$$\log p(\theta|y, X) = \log \mathcal{N}(y|\Phi^T\theta, \Sigma) + \log \mathcal{N}(\theta|0, \Gamma) + \text{const.} \tag{S.223}$$

$$= -\frac{1}{2} \left\|y - \Phi^T\theta\right\|_{\Sigma}^2 - \frac{1}{2}\theta^T\Gamma^{-1}\theta + \text{const.} \tag{S.224}$$

$$= -\frac{1}{2} \left(-2\theta^T\Phi\Sigma^{-1}y + \theta^T\Phi\Sigma^{-1}\Phi^T\theta\right) - \frac{1}{2}\theta^T\Gamma^{-1}\theta + \text{const.} \tag{S.225}$$

$$= -\frac{1}{2}[\theta^T \underbrace{(\Phi\Sigma^{-1}\Phi^T + \Gamma^{-1})}_{A}\theta - 2\theta^T \underbrace{\Phi\Sigma^{-1}y}_{b}] + \text{const.} \tag{S.226}$$

Next, using the above A and b, we complete the square (see Problem 1.10),

$$\log p(\theta|y, X) = -\frac{1}{2}(\theta - A^{-1}b)^T A(\theta - A^{-1}b) + \text{const.} \tag{S.227}$$

$$= -\frac{1}{2} \left\|\theta - \hat{\mu}_\theta\right\|_{\hat{\Sigma}_\theta}^2 + \text{const.} \tag{S.228}$$

where again constant terms are ignored, and we define

$$\hat{\mu}_\theta = A^{-1}b = (\Phi\Sigma^{-1}\Phi^T + \Gamma^{-1})^{-1}\Phi\Sigma^{-1}y, \tag{S.229}$$

$$\hat{\Sigma}_\theta = A^{-1} = (\Phi\Sigma^{-1}\Phi^T + \Gamma^{-1})^{-1}. \tag{S.230}$$

Finally, note that the log-posterior in (S.228) is of the same form of a Gaussian for θ. Hence the posterior is Gaussian,

$$p(\theta|y, X) = \mathcal{N}(\theta|\hat{\mu}_\theta, \hat{\Sigma}_\theta) \tag{S.231}$$

(b) The MAP solution is the mean of the Gaussian (i.e., the θ with largest likelihood),

$$\hat{\theta}_{MAP} = \underset{\theta}{\operatorname{argmax}} \mathcal{N}(\theta|\hat{\mu}_\theta, \hat{\Sigma}_\theta) \tag{S.232}$$

$$= \hat{\mu}_\theta = (\Phi\Sigma^{-1}\Phi^T + \Gamma^{-1})^{-1}\Phi\Sigma^{-1}y. \tag{S.233}$$

The $\hat{\theta}_{MAP}$ is similar to the weighted least-squares estimate, but has an additional term Γ^{-1}. When $\Gamma^{-1} = 0$, then $\hat{\theta}_{MAP}$ is the same as the weighted least squares solution. For non-zero values of Γ, then the term serves to regularize the covariance matrix $\Phi\Sigma^{-1}\Phi^T$, which might not be strictly positive definite or nearly singular. E.g., if $\Gamma = I$ is a diagonal matrix, then this would guarantee that the matrix inverse of $(\Phi\Sigma^{-1}\Phi^T + \Gamma^{-1})$ can always be performed.

(c) Substituting for $\Gamma = \alpha I$ and $\Sigma = \sigma^2 I$ in $\hat{\theta}_{MAP}$,

$$\hat{\theta} = (\Phi(\tfrac{1}{\sigma^2}I)\Phi^T + \tfrac{1}{\alpha}I)^{-1}\Phi\tfrac{1}{\sigma^2}Iy = (\Phi\Phi^T + \tfrac{\sigma^2}{\alpha}I)^{-1}\Phi y = (\Phi\Phi^T + \lambda I)^{-1}\Phi y, \tag{S.234}$$

where $\lambda = \frac{\sigma^2}{\alpha} \geq 0$.

To solve the regularized least-squares problem, first consider the objective function

$$R = \left\|y - \Phi^T\theta\right\|^2 + \lambda \left\|\theta\right\|^2 = (y - \Phi^T\theta)^T(y - \Phi^T\theta) + \lambda\theta^T\theta \tag{S.235}$$

$$= y^Ty - 2y^T\Phi^T\theta + \theta^T\Phi\Phi^T\theta + \lambda\theta^T\theta \tag{S.236}$$

$$= y^Ty - 2y^T\Phi^T\theta + \theta^T(\Phi\Phi^T + \lambda I)\theta. \tag{S.237}$$

Taking the derivative and setting to 0,

$$\frac{\partial R}{\partial \theta} = -2\Phi y + 2(\Phi\Phi^T + \lambda I)\theta = 0 \quad \Rightarrow \quad (\Phi\Phi^T + \lambda I)\theta = \Phi y \tag{S.238}$$

$$\Rightarrow \quad \theta = (\Phi\Phi^T + \lambda I)^{-1}\Phi y \tag{S.239}$$

Hence the regularized least-squares estimate (aka ridge regression) is equivalent to the above MAP estimate for Bayesian regression with Gaussian noise and prior with isotropic covariances.

(d) Substituting for $\Gamma = \alpha I$ and $\Sigma = \sigma^2 I$, the posterior mean and covariance are

$$\hat{\mu}_\theta = (\Phi\Phi^T + \tfrac{\sigma^2}{\alpha}I)^{-1}\Phi y, \qquad \hat{\Sigma}_\theta = (\tfrac{1}{\sigma^2}\Phi\Phi^T + \tfrac{1}{\alpha}I)^{-1}. \tag{S.240}$$

Here are the various cases of interest:

- Setting $\alpha \to 0$ corresponds to setting a very strong prior at $\theta = 0$, since the covariance of the prior will be $\Gamma = 0$. The term $\frac{1}{\alpha}I$ is a diagonal matrix with very large entries, and its inverse is a matrix that is zero. As a result, the posterior of θ is equivalent to the prior, $\hat{\mu}_\theta = 0$ and $\hat{\Sigma}_\theta = 0$.
- Setting $\alpha \to \infty$ yields a very weak prior since the covariance Γ is very large. As a result, the the $\frac{1}{\alpha}I$ term vanishes, leaving an estimate equivalent to ordinary least squares, $\hat{\mu}_\theta = (\Phi\Phi^T)^{-1}\Phi y$ and $\hat{\Sigma}_\theta = (\frac{1}{\sigma^2}\Phi\Phi^T)^{-1}$.
- When $\sigma^2 \to 0$, then this means there is no observation noise. As a result, the prior is ignored and the data term dominates the mean $\hat{\mu}_\theta = (\Phi\Phi^T)^{-1}\Phi y$, resulting in the ordinary least square estimate. The posterior covariance is $\hat{\Sigma}_\theta = 0$, since there is no uncertainty in the observations.

- When $\sigma^2 \to \infty$ this corresponds to observations being very very noisy. As a result, the data term is ignored and the prior dominates, resulting a posterior equivalent to the prior, $\hat{\mu}_\theta = 0$ and $\hat{\Sigma}_\theta = \alpha I$.

(e) Given a novel input x_*, we are interested in the function value f_* conditioned on the data $\{y, X, x_*\}$. Note that the posterior $\theta|y, X$ is a Gaussian random variable. Hence, when conditioning on the data, $f_* = \phi(x_*)^T\theta$ is a linear transformation of a Gaussian random variable, which is also Gaussian. Using the properties in Problem 1.1, which give the mean and variance of the transformed Gaussian, we have

$$p(f_*|y, X, x*) = \mathcal{N}(f_*|\hat{\mu}_*, \hat{\sigma}_*^2), \quad \hat{\mu}_* = \phi(x_*)^T\hat{\mu}_\theta, \quad \hat{\sigma}_*^2 = \phi(x_*)^T\hat{\Sigma}_\theta\phi(x_*). \tag{S.241}$$

Finally, for the predicted y_*, we have obtain the distribution by integrating over f_*,

$$p(y_*|y, X, x_*) = \int p(y_*|f_*)p(f_*|y, X, x_*)df_* = \int \mathcal{N}(y_*|f_*, \sigma^2)\mathcal{N}(f_*|\hat{\mu}_*, \hat{\sigma}_*^2)df_* \tag{S.242}$$

$$= \int \mathcal{N}(f_*|y_*, \sigma^2)\mathcal{N}(f_*|\hat{\mu}_*, \hat{\sigma}_*^2)df_* = \mathcal{N}(y_*|\hat{\mu}_*, \hat{\sigma}_*^2 + \sigma^2). \tag{S.243}$$

where the last line uses Problem 1.9 to calculate the integral (correlation between Gaussian distributions).

.....

Problem 3.12 L1-regularized least-squares (LASSO)

(a) The L1-regularized least-squares objective function is

$$E = \frac{1}{2}\left\|y - \Phi^T\theta\right\|^2 + \lambda\sum_{i=1}^{D}|\theta_i|. \tag{S.244}$$

The first term (data term) is the squared-error as in ordinary least squares. The second term (regularization term) is the absolute value of the parameter values. In contrast, regularized least squares (ridge regression) uses the norm of the parameter $\|\theta\|^2$. The objective function E is equivalent to the negative log-likelihood, so the likelihood takes the form of

$$\ell \propto e^{-E} = e^{-\left\|y-\Phi^T\theta\right\|^2 - \lambda\sum_{i=1}^{D}|\theta_i|} = e^{-\sum_{i=1}^{N}(y_i-\phi(x_i)^T\theta)^2 - \lambda\sum_{i=1}^{D}|\theta_i|} \tag{S.245}$$

$$= \left[\prod_{i=1}^{N} e^{-(y_i-\phi(x_i)^T\theta)^2}\right]\left[\prod_{i=1}^{D} e^{-\lambda|\theta_i|}\right]. \tag{S.246}$$

The left term is the data likelihood, which is equivalent to a Gaussian (as in ordinary least squares). The right term is the prior on θ, and takes the form of a Laplacian on each parameter value. Hence, the probabilistic interpretation of L1-regularized least squares is to find the MAP estimate of the parameters θ, using the model

$$y = f(x;\theta) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \quad \theta_i \sim \text{Laplace}(\lambda) = \tfrac{1}{2\lambda}e^{-\frac{|\theta_i|}{\lambda}}. \tag{S.247}$$

(b) In the equivalent optimization problem, we rewrite θ_i as the difference between two positive values, $\theta_i = \theta_i^+ - \theta_i^-$, where $\theta_i^+ \geq 0$ and $\theta_i^- \geq 0$. Note that when $\theta_i^+ = 0$ and $\theta_i^- > 0$, and vice versa ($\theta_i^+ > 0$, $\theta_i^- = 0$), then the absolute value can be rewritten as $|\theta_i^+ - \theta_i^-| = (\theta_i^+ + \theta_i^-)$. Finally, at the optimum indeed one of these two conditions hold, $(\theta_i^+ = 0, \theta_i^- > 0)$ or $(\theta_i^+ > 0, \theta_i^- = 0)$. If it were not the case, i.e., $(\theta_i^+ > 0, \theta_i^- > 0)$, then the term $(\theta_i^+ + \theta_i^-)$ could be further reduced by subtracting $\min(\theta_i^+, \theta_i^-)$ from θ_i^+ and θ_i^-. This would reduce the regularization term $(\theta_i^+ + \theta_i^-)$, but not affect the data term since the data term only depends on the difference $(\theta_i^+ - \theta_i^-)$.

(c) Let $\mathbf{x} = \begin{bmatrix} \theta^+ \\ \theta^- \end{bmatrix}$. The objective function is

$$E = \frac{1}{2} \left\| y - \Phi^T(\theta^+ - \theta^-) \right\|^2 + \lambda \sum_i (\theta_i^+ + \theta_i^-) \tag{S.248}$$

$$= \frac{1}{2} \left\| y - \left[\Phi^T, -\Phi^T\right] \mathbf{x} \right\|^2 + \lambda \mathbf{1}^T \mathbf{x} \tag{S.249}$$

$$= \frac{1}{2} y^T y - y^T \left[\Phi^T, -\Phi^T\right] \mathbf{x} + \frac{1}{2} \mathbf{x}^T \begin{bmatrix} \Phi \\ -\Phi \end{bmatrix} \left[\Phi^T, -\Phi^T\right] \mathbf{x} + \lambda \mathbf{1}^T \mathbf{x} \tag{S.250}$$

$$\propto \frac{1}{2} \mathbf{x}^T \underbrace{\begin{bmatrix} \Phi\Phi^T & -\Phi\Phi^T \\ -\Phi\Phi^T & \Phi\Phi^T \end{bmatrix}}_{\mathbf{H}} \mathbf{x} + \underbrace{\left(\lambda \mathbf{1} - \begin{bmatrix} \Phi y \\ -\Phi y \end{bmatrix} \right)^T}_{\mathbf{f}} \mathbf{x}, \tag{S.251}$$

where constant terms that do not affect the minimization are dropped.

Figure 8 shows an example of cubic polynomial regression using least-squares, Bayesian regression, and LASSO.

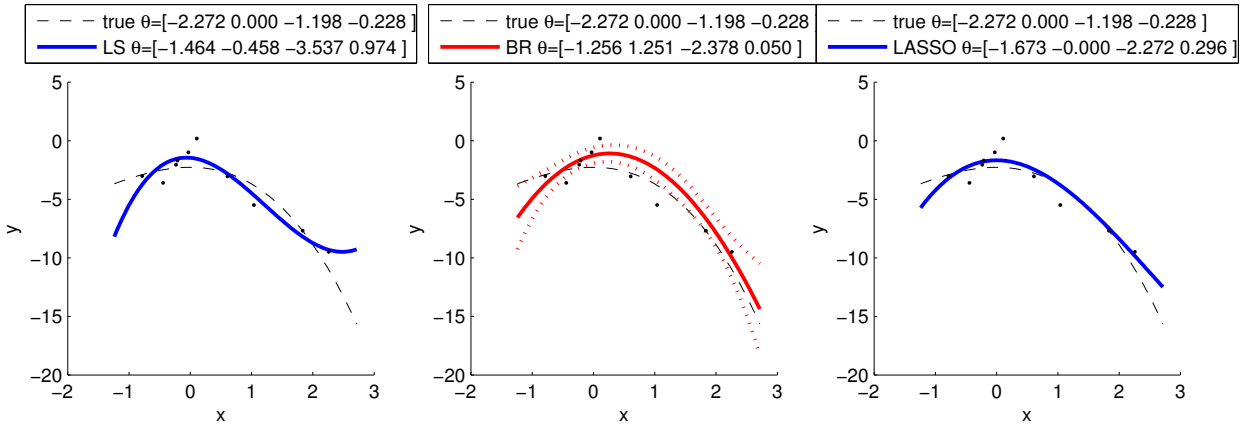

Figure 8: Cubic polynomial regression using (left) least-squares, (middle) Bayesian regression, (right) LASSO. The true function is the dashed line. For Bayesian regression, the dotted-lines show the 2 standard-deviations around the mean. Note that LASSO can find that the linear term θ_1 is 0.

. . . . . .

Problem 4.12 Lagrange multipliers and equality constraints

(a) The Lagrangian is

$$L(\pi, \lambda) = \sum_{j=1}^{K} z_j \log \pi_j + \lambda \left(\sum_{j=1}^{K} \pi_j - 1 \right). \tag{S.252}$$

Taking the derivatives and setting to zero gives

$$\frac{\partial L}{\partial \lambda} = \sum_{j=1}^{K} \pi_j - 1 = 0 \quad \Rightarrow \quad \sum_{j=1}^{K} \pi_j = 1, \tag{S.253}$$

$$\frac{\partial L}{\partial \pi_j} = \frac{z_j}{\pi_j} + \lambda = 0 \quad \Rightarrow \quad z_j + \pi_j \lambda = 0. \tag{S.254}$$

Taking (S.254) and summing over j,

$$\sum_{j=1}^{K} (z_j + \pi_j \lambda) = 0 \quad \Rightarrow \quad \lambda \sum_{j=1}^{K} \pi_j = -\sum_{j=1}^{K} z_j \quad \Rightarrow \quad \lambda = -\sum_{j=1}^{K} z_j, \tag{S.255}$$

since $\sum_{k=1}^{K} \pi_k = 1$. Finally substituting λ into (S.254) (and changing the index from j to k to avoid confusion),

$$z_j - \pi_j \sum_{k=1}^{K} z_j = 0 \quad \Rightarrow \quad \pi_j = \frac{z_j}{\sum_{k=1}^{K} z_k}. \tag{S.256}$$

(b) The Lagrangian is

$$L(\pi, \lambda) = \sum_{j=1}^{K} \pi_j (z_j - \log \pi_j) + \lambda \left(\sum_{j=1}^{K} \pi_j - 1 \right). \tag{S.257}$$

Taking the derivatives and setting to zero gives

$$\frac{\partial L}{\partial \lambda} = \sum_{j=1}^{K} \pi_j - 1 = 0 \quad \Rightarrow \quad \sum_{j=1}^{K} \pi_j = 1, \tag{S.258}$$

$$\frac{\partial L}{\partial \pi_j} = z_j - \log \pi_j - \frac{\pi_j}{\pi_j} + \lambda = 0 \quad \Rightarrow \quad \frac{1}{\pi_j} e^{z_j - 1} e^{\lambda} = 0 \quad \Rightarrow \quad \pi_j e^{-\lambda} = e^{z_j - 1} \tag{S.259}$$

Summing over j and noting that $\sum_{j=1}^{K} \pi_j = 1$ gives

$$e^{-\lambda} = \sum_{j=1}^{K} e^{z_j - 1}. \tag{S.260}$$

Finally, substituting (S.260) back into (S.259),

$$\pi_j \sum_{k=1}^{K} e^{z_k - 1} = e^{z_j - 1} \quad \Rightarrow \quad \pi_j = \frac{e^{z_j - 1}}{\sum_{k=1}^{K} e^{z_k - 1}} = \frac{e^{z_j}}{\sum_{j=1}^{K} e^{z_k}} \tag{S.261}$$

.....

Problem 4.6 Mixture of exponentials

The derivation is similar to that of Problem 4.5 (mixture of Poissons). The only difference is the observation likelihood is now an exponential density rather than a Poisson. In particular, the $\mathcal{Q}$ function is

$$Q(\theta;\hat{\theta}) = \mathbb{E}_{Z|X,\hat{\theta}}\left[\log p(X,Z|\theta)\right] = \sum_{i=1}^{n}\sum_{j=1}^{K} \hat{z}_{ij}\log\pi_j + \hat{z}_{ij}\log p(x_i|z_i=j), \tag{S.262}$$

where the soft assignments are

$$\hat{z}_{ij} = \mathbb{E}_{Z|X,\hat{\theta}}[z_{ij}] = \frac{\pi_j p(x_i|z_i=j)}{\sum_{k=1}^{K}\pi_k p(x_i|z_i=k)} = p(z_i=j|x_i,\hat{\theta}). \tag{S.263}$$

For the M-step, the mixture weights are updated as before,

$$\hat{\pi}_j = \frac{1}{n}\sum_{i=1}^{n}\hat{z}_{ij} = \frac{\hat{n}_j}{n}, \tag{S.264}$$

where $\hat{n}_j = \sum_{i=1}^{n}\hat{z}_{ij}$ is the (soft) number of samples assigned to component j.

Finally, for the exponential parameter λ_j, we collect the terms of the $\mathcal{Q}$ function that depend on λ_j,

$$\ell_j = \sum_{i=1}^{n}\hat{z}_{ij}\log p(x_i|z_i=j) = \sum_{i=1}^{n}\hat{z}_{ij}(\log\lambda_j - \lambda_j x_i). \tag{S.265}$$

Taking the derivative and setting to 0 gives

$$\frac{\partial\ell_j}{\partial\lambda_j} = \sum_{i=1}^{n}\hat{z}_{ij}(\lambda_j^{-1} - x_i) = 0 \quad\Rightarrow\quad \lambda_j^{-1}\sum_{i=1}^{n}\hat{z}_{ij} = \sum_{i=1}^{n}\hat{z}_{ij}x_i \quad\Rightarrow\quad \lambda_j^{-1} = \frac{\sum_{i=1}^{n}\hat{z}_{ij}x_i}{\sum_{i=1}^{n}\hat{z}_{ij}}. \tag{S.266}$$

.....

Problem 5.1 Bias and variance of the kernel density estimator

(a) To calculate the bias and variance of the kernel density estimator $\hat{p}(x) = \frac{1}{n}\sum_{i=1}^{n}\tilde{k}(x-x_i)$, we suppose that the samples $X = \{x_i\}_{i=1}^{n}$ are distributed according to the true distribution $p(x)$, i.e., $x_i \sim p(x), \forall i$. The mean of the estimator $\hat{p}(x)$ is

$$\mathbb{E}_X[\hat{p}(x)] = \mathbb{E}_X\left[\frac{1}{n}\sum_{i=1}^{n}\tilde{k}(x-x_i)\right] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{x_i}\left[\tilde{k}(x-x_i)\right] \tag{S.267}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\int p(x_i)\tilde{k}(x-x_i)dx_i = \int p(z)\tilde{k}(x-z)dz = p(x)*\tilde{k}(x), \tag{S.268}$$

where (S.268) follows from each term in the sum being the same, and $*$ is the convolution operator. The mean of estimator is the true distribution convolved with the kernel. In other words, it is a "blurred" or "smoothed" version of the true distribution.

(b) The variance of the estimator is

$$\text{var}_X(\hat{p}(x)) = \text{var}\left(\frac{1}{n}\sum_{i=1}^{n}\tilde{k}(x-x_i)\right) = \frac{1}{n^2}\text{var}\left(\sum_{i=1}^{n}\tilde{k}(x-x_i)\right) \tag{S.269}$$

$$= \frac{1}{n^2}\sum_{i=1}^{n}\text{var}\left(\tilde{k}(x-z)\right) = \frac{1}{n}\text{var}\left(\tilde{k}(x-x_i)\right), \tag{S.270}$$

which follows from $\{x_i\}$ being independent distributions, and hence the variance of the sum is the sum of the variances (see Problem 1.4), and also identical distributions. Noting that $\text{var}(x) = \mathbb{E}[x^2] - \mathbb{E}[x]^2$ and thus $\text{var}(x) \leq \mathbb{E}[x^2]$, then we can place an upper-bound on the variance,

$$\text{var}_X(\hat{p}(x)) \leq \frac{1}{n}\mathbb{E}[\tilde{k}(x-z)^2] = \frac{1}{n}\int \frac{1}{h^d}k\left(\frac{x-z}{h}\right)\tilde{k}(x-z)p(z)dz \tag{S.271}$$

$$\leq \frac{1}{nh^d}\left(\max_x k(x)\right)\int \tilde{k}(x-z)p(z)dz = \frac{1}{nh^d}\left(\max_x k(x)\right)\mathbb{E}[\hat{p}(x)], \tag{S.272}$$

where the last line follows from $k(\frac{x-z}{h}) \leq \max_x k(x)$, i.e., the kernel is upper-bounded by its maximum value.

Figure 9 plots the mean and variance of the KDE for different bandwidths.

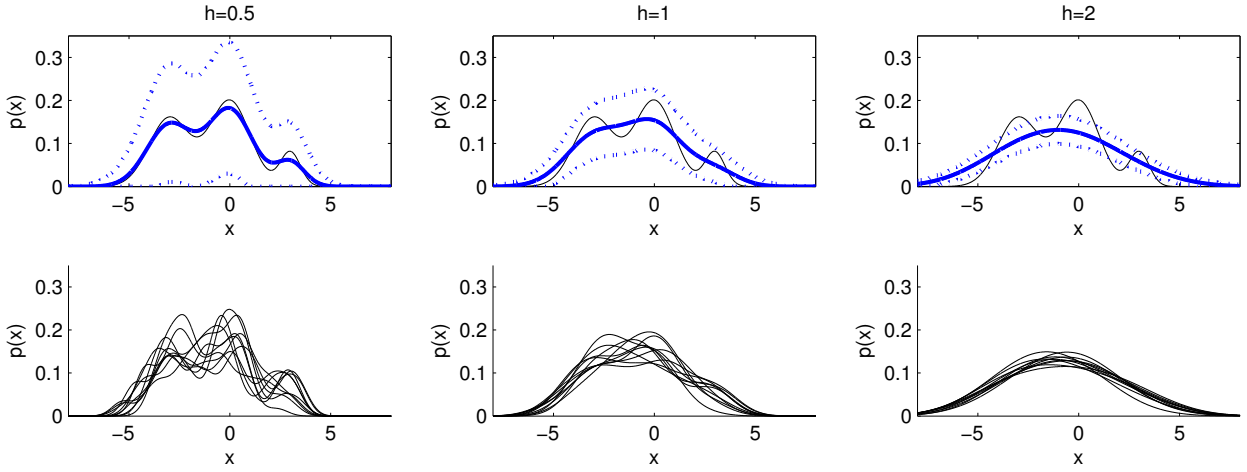

Figure 9: Mean and variance of kernel density estimator. Top row: the true density function (thin line) and the estimator mean (thick line) and 2 standard deviations around the mean (dotted line) for different bandwidths h. Bottom row: Examples of the estimate $\hat{p}(x)$ using 10 different sets of 50 samples drawn from the true density $p(x)$. When the bandwidth is small ($h = 0.5$), the estimates are significantly different for each sample set (i.e., high variance), but the estimator mean is close to the true density. When the bandwidth is large ($h = 2$), the estimates are consistent with each other (i.e., low variance), but the estimator mean is far from the true density.

. . . . . .

Problem 6.2 BDR for regression

(a) For a given x, the conditional risk function for the squared loss function is

$$R(x) = \mathbb{E}_{y|x}[L(g(x), y)] = \int p(y|x)(g(x) - y)^2 dy \tag{S.273}$$

$$= \int p(y|x)(g(x)^2 - 2yg(x) + y^2)dy = g(x)^2 - 2g(x)\mathbb{E}[y|x] + \mathbb{E}[y^2|x]. \tag{S.274}$$

We wish to minimize the risk w.r.t. the choice of $g(x)$. Hence, taking the derivative w.r.t. $g(x)$ and setting to 0,

$$\frac{\partial R(x)}{\partial g(x)} = 2g(x) - 2\mathbb{E}[y|x] = 0 \quad \Rightarrow \quad g^*(x) = \mathbb{E}[y|x]. \tag{S.275}$$

(b) The Minkowski loss function is plotted in Figure 10. For $q = 1$ it corresponds to the absolute loss (L1-norm), while for $q = 2$ it is the squared loss (L2-norm). As $q \to 0$, it becomes the 0-1 loss function, $L_0 = \begin{cases} 0, & g(x) = y \\ 1, & \text{otherwise} \end{cases}$.

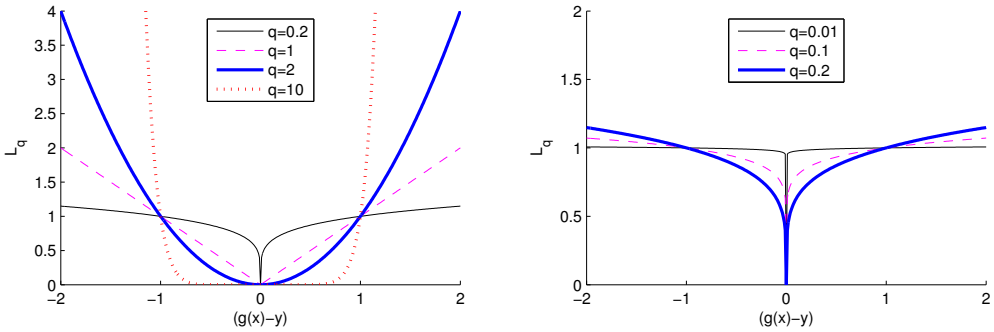

Figure 10: Minkowski loss function $L_q = |g(x) - y|^q$ for different values of q.

(c) The conditional risk function for the $q = 1$ is

$$R(x) = \int p(y|x)\,|g - y|\,dy \tag{S.276}$$

$$= \int_{-\infty}^{g} p(y|x)(y - g)dy + \int_{g}^{\infty} p(y|x)(g - y)dy \tag{S.277}$$

$$= E_-[y] - gE_-[1] + gE_+[1] - E_+[y], \tag{S.278}$$

where we have defined the following partial expectation terms,

$$E_-[y] = \int_{-\infty}^{g} p(y|x)ydy, \quad E_+[y] = \int_{g}^{\infty} p(y|x)ydy, \tag{S.279}$$

$$E_-[1] = \int_{-\infty}^{g} p(y|x)dy, \quad E_+[1] = \int_{g}^{\infty} p(y|x)dy. \tag{S.280}$$

To minimize the risk we need to take the derivative w.r.t. g. However, g also appears in the limits of the integral. Recall the "fundamental theorem of calculus",

$$\frac{\partial}{\partial x}\int_a^x f(t)dt = f(x) \quad \text{and} \quad \frac{\partial}{\partial x}\int_x^a f(t)dt = -f(x). \tag{S.281}$$

Hence, the derivatives of the partial expectations are

$$\frac{\partial}{\partial g}E_-[y] = p(g|x)g, \quad \frac{\partial}{\partial g}E_+[y] = -p(g|x)g, \tag{S.282}$$

$$\frac{\partial}{\partial g}E_-[1] = p(g|x), \quad \frac{\partial}{\partial g}E_+[1] = -p(g|x). \tag{S.283}$$

Finally, taking the derivative of $R(x)$ and setting to 0,

$$\frac{\partial R(x)}{\partial g} = p(g|x)g - (E_-[1] + gp(g|x)) + (E_+[1] - gp(g|x)) - (-p(g|x)g) = 0 \tag{S.284}$$

$$\Rightarrow \quad -E_-[1] + E_+[1] = 0 \tag{S.285}$$

Hence, at the minimum we have the condition on g that

$$E_-[1] = E_+[1] \tag{S.286}$$

$$\int_{-\infty}^{g} p(y|x)dy = \int_g^{\infty} p(y|x)dy = \frac{1}{2}, \tag{S.287}$$

since the density integrates to 1. The optimal g is the value that splits the posterior distribution $p(y|x)$ in half, i.e., the median of $p(y|x)$.

.....

Problem 6.6 Gaussian classifier with common covariance

(a) The BDR for a classifier with 0-1 loss function is

$$g(x) = \underset{j}{\text{argmax}} \underbrace{\log p(x|j) + \log p(j)}_{g_j(x)}, \tag{S.288}$$

where the decision function for each class j is

$$g_j(x) = \log p(x|j) + \log p(j) \tag{S.289}$$

$$= -\frac{1}{2}\|x-\mu_j\|_\Sigma^2 - \frac{1}{2}\log|\Sigma| - \frac{d}{2}\log 2\pi + \log \pi_j \tag{S.290}$$

$$\propto -\frac{1}{2}\|x-\mu_j\|_\Sigma^2 + \log \pi_j \tag{S.291}$$

$$= -\frac{1}{2}(x-\mu_j)^T\Sigma^{-1}(x-\mu_j) + \log \pi_j \tag{S.292}$$

$$= -\frac{1}{2}(x^T\Sigma^{-1}x - 2\mu_j^T\Sigma^{-1}x + \mu_j^T\Sigma^{-1}\mu_j) + \log \pi_j \tag{S.293}$$

$$\propto \underbrace{\mu_j^T\Sigma^{-1}}_{w_j^T}x + \underbrace{-\frac{1}{2}\mu_j^T\Sigma^{-1}\mu_j + \log \pi_j}_{b_j} = w_j^T x + b_j \tag{S.294}$$

(b) The boundary between two classes i and j occurs where the two decision functions are equal,

$$g_i(x) = g_j(x) \tag{S.295}$$
$$w_i^T x + b_i = w_j^T x + b_j \tag{S.296}$$
$$\Rightarrow \quad (\underbrace{w_i - w_j}_{w})^T x + \underbrace{(b_i - b_j)}_{b} = 0 = w^T x + b, \tag{S.297}$$

where

$$w = (w_i - w_j) = \Sigma^{-1}(\mu_i - \mu_j), \tag{S.298}$$
$$b = (b_i - b_j) = \left[-\frac{1}{2}\mu_i^T \Sigma^{-1}\mu_i + \log \pi_i\right] - \left[\frac{1}{2}\mu_j^T \Sigma^{-1}\mu_j + \log \pi_j\right] \tag{S.299}$$
$$= -\frac{1}{2}(\mu_i + \mu_j)^T \Sigma^{-1}(\mu_i - \mu_j) + \log \frac{\pi_i}{\pi_j}. \tag{S.300}$$

The last line follows from the identity $a^T C a - b^T C b = (a+b)^T C (a-b)$.

(c) To rewrite the hyperplane in the form $w^T(x - x_0) = 0$, we note that $w^T x - w^T x_0 = 0$, and hence we must find an x_0 such that $b = -w^T x_0$.

$$b = -w^T x_0 \tag{S.301}$$
$$-\frac{1}{2}(\mu_i + \mu_j)^T \Sigma^{-1}(\mu_i - \mu_j) + \log \frac{\pi_i}{\pi_j} = -(\mu_i - \mu_j)^T \Sigma^{-1} x_0 \tag{S.302}$$
$$-\frac{1}{2}(\mu_i + \mu_j)^T \Sigma^{-1}(\mu_i - \mu_j) + \frac{\|\mu_i - \mu_j\|_\Sigma^2}{\|\mu_i - \mu_j\|_\Sigma^2} \log \frac{\pi_i}{\pi_j} = -(\mu_i - \mu_j)^T \Sigma^{-1} x_0 \tag{S.303}$$
$$-(\mu_i - \mu_j)^T \Sigma^{-1} \left[\frac{1}{2}(\mu_i + \mu_j) - \frac{(\mu_i - \mu_j)}{\|\mu_i - \mu_j\|_\Sigma^2} \log \frac{\pi_i}{\pi_j}\right] = -(\mu_i - \mu_j)^T \Sigma^{-1} x_0 \tag{S.304}$$

Hence by inspection,

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{(\mu_i - \mu_j)}{\|\mu_i - \mu_j\|_\Sigma^2} \log \frac{\pi_i}{\pi_j}. \tag{S.305}$$

.....

Problem 7.4 PCA implementation using SVD

(a) $\bar{X}$ is the matrix of mean-subtracted points,

$$\bar{X} = [x_1 - \mu, \cdots, x_n - \mu] = X - [\mu, \cdots \mu] = X - \mu \mathbf{1}^T \tag{S.306}$$
$$= X - \left(\frac{1}{n}\sum_{i=1}^n x_i\right)\mathbf{1}^T = X - \left(\frac{1}{n} X \mathbf{1}\right)\mathbf{1}^T = X(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T). \tag{S.307}$$

(b) The covariance can be written as the outer-product of $\bar{X}$,

$$\Sigma = \frac{1}{n}\bar{X}\bar{X}^T. \tag{S.308}$$

Substituting for the SVD of $\bar{X}$,

$$\Sigma = \frac{1}{n} USV^T (USV^T)^T = \frac{1}{n} USV^T VSU^T = \frac{1}{n} US^2 U^T, \tag{S.309}$$

since $V^T V = I$. Finally,

$$\Sigma = U \left(\frac{1}{n} S^2 \right) U^T. \tag{S.310}$$

Note that S^2 is a diagonal matrix and also $U^T U = I$. Hence U is a matrix of eigenvectors and $\frac{1}{n} S^2$ is a diagonal matrix of eigenvalues. In particular, the i-th eigen-pair is $\{u_i, \frac{s_i^2}{n}\}$.

(c) The PCA algorithm using SVD is

1. Center the data: $\bar{X} = X(I - \frac{1}{n} \mathbf{1}\mathbf{1}^T)$.
2. Calculate the SVD: $\bar{X} = USV^T$.
3. Get the top K singular values, $s_1 \geq s_2 \geq \cdots \geq s_K$.
4. Form the PCA basis matrix, $\Phi = [u_1, \cdots, u_K]$.
5. Project data: $z = \Phi^T x$.

. . . . . .

Problem 7.6 Fisher's linear discriminant

(a) Fixing the denominator of the ratio $J(w)$ yields the equivalent optimization problem,

$$w^* = \underset{w}{\operatorname{argmax}}\, w^T S_B w \quad \text{s.t. } w^T S_W w = 1. \tag{S.311}$$

The Lagrangian is

$$L(w, \lambda) = w^T S_B w - \lambda (w^T S_W w - 1). \tag{S.312}$$

(b) Taking the derivative w.r.t. w and setting to 0,

$$\frac{\partial L}{\partial w} = 2 S_B w - 2 \lambda S_W w = 0 \quad \Rightarrow \quad S_B w = \lambda S_W w. \tag{S.313}$$

This is a generalized eigenvalue problem.

(c) Assuming that S_W is invertible,

$$\lambda w = S_W^{-1} S_B w \tag{S.314}$$

$$= S_W^{-1} (\mu_1 - \mu_2) \underbrace{(\mu_1 - \mu_2)^T w}_{\text{scalar}} \tag{S.315}$$

Note that the scale of w does not affect the ratio $J(w) = \frac{w^T S_B w}{w^T S_W w}$. Hence, we only need the direction of w. Thus, we can drop the scalar terms, λ and $(\mu_1 - \mu_2)^T w$, and the solution is

$$w^* = S_W^{-1} (\mu_1 - \mu_2). \tag{S.316}$$

. . . . . .

Problem 8.6 Newton-Raphson Method

(a) The derivative $f'(x^{(i)})$ of $f(x)$ at $x^{(i)}$ is equal to the change in the function divided by the change in the input,

$$f'(x^{(i)}) = \frac{\Delta f}{\Delta x} \tag{S.317}$$

To find the zero-crossing point, we substitute the current point $(x^{(i)}, f(x^{(i)}))$ and the desired zero-crossing point $(x^{(i+1)}, 0)$,

$$f'(x^{(i)}) = \frac{f(x^{(i)}) - 0}{x^{(i)} - x^{(i+1)}}. \tag{S.318}$$

Solving for $x^{(i+1)}$ gives the iteration,

$$x^{(i+1)} = x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})}. \tag{S.319}$$

(b) Figure 11 shows the Newton-Raphson method on the polynomial $f(x) = x^3 - 2x + 2$ when starting at $x^{(1)} = 0$.

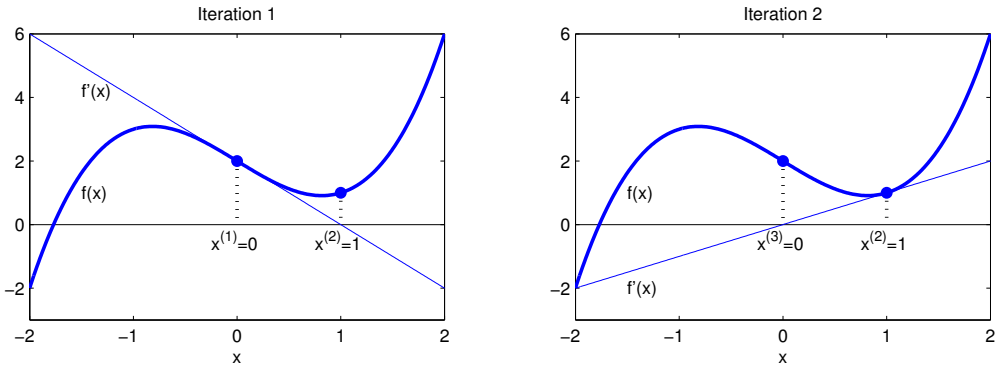

Figure 11: Newton-Raphson method for the polynomial $f(x) = x^3 - 2x + 2$ when starting at $x^{(1)} = 0$. In the first iteration, the next point is calculated as $x^{(2)} = 1$. In the second iteration, the next point is calculated as $x^{(3)} = 0$. Hence, the iteration repeats indefinitely and fails to converge.

(c) The condition of the optimum of $g(x)$ is $g'(x) = 0$. Hence, we want to find the zero-crossing of $f(x) = g'(x)$, yielding the iteration,

$$x^{(i+1)} = x^{(i)} - \frac{1}{g''(x^{(i)})} g'(x^{(i)}) \tag{S.320}$$

The iteration is similar to gradient descent/ascent in that a scaled version of the gradient is added to the current point in each iteration. The main difference is that the scale is chosen adaptively using the second derivative $g''(x^{(i)})$.

(d) Let $h(x)$ be the Taylor approximation of $g(x)$ around the point $a = x^{(i)}$,

$$h(x) = g(a) + g'(a)(x - a) + \frac{g''(a)}{2}(x - a)^2 \tag{S.321}$$

To find the optimum of $h(x)$, we take the derivative and set to 0,

$$\frac{\partial h(x)}{\partial x} = g'(a) + g''(a)(x-a) = 0 \quad \Rightarrow \quad x - a = -\frac{g'(a)}{g''(a)} \tag{S.322}$$

$$\Rightarrow \quad x^* = a - \frac{g'(a)}{g''(a)}. \tag{S.323}$$

This is equivalent to the Newton-Raphson iteration in (S.320). Hence, using the Newton-Raphson method for finding the optimum of a function $g(x)$ can be interpreted as iteratively forming a quadratic Taylor approximation $h(x)$ at the current point, and then moving to the optimum of the approximation $h(x)$.

.....

Problem 8.4 Regularized Logistic Regression: MAP framework

(a) The MAP objective function is

$$\ell = \log p(y|X,w) + \log p(w) = \sum_{i=1}^{n} \log p(y_i|x_i,w) + \log p(w) \tag{S.324}$$

$$= \sum_{i=1}^{n} \left[y_i \log \pi_i + (1-y_i)\log(1-\pi_i)\right] - \frac{1}{2} w^T \Gamma w - \frac{1}{2}\log|\Gamma| - \frac{d}{2}\log 2\pi. \tag{S.325}$$

Dropping terms that are constant, the MAP solution is equivalent to

$$w^* = \underset{w}{\operatorname{argmax}}\, \ell = \underset{w}{\operatorname{argmin}}\, -\ell \tag{S.326}$$

$$= \underset{w}{\operatorname{argmin}} \underbrace{-\sum_{i=1}^{n} \left[y_i \log \pi_i + (1-y_i)\log(1-\pi_i)\right]}_{E(w)} + \frac{1}{2} w^T \Gamma w \tag{S.327}$$

$$= \underset{w}{\operatorname{argmin}}\, \hat{E}(w), \quad \hat{E}(w) = E(w) + \frac{1}{2} w^T \Gamma w. \tag{S.328}$$

(b) First let's look at the gradient of $E(w)$,

$$\nabla E(w) = \frac{\partial}{\partial w} - \sum_{i=1}^{n} \left[y_i \log \pi_i + (1-y_i)\log(1-\pi_i)\right] \tag{S.329}$$

$$= -\sum_{i=1}^{n} \left[y_i \frac{\partial}{\partial w} \log \pi_i + (1-y_i) \frac{\partial}{\partial w} \log(1-\pi_i) \right] \tag{S.330}$$

$$= -\sum_{i=1}^{n} \left[y_i \frac{1}{\pi_i} \frac{\partial \pi_i}{\partial w} + (1-y_i) \frac{-1}{(1-\pi_i)} \frac{\partial \pi_i}{\partial w} \right]. \tag{S.331}$$

Using Problem 8.1a and the chain rule, we obtain the derivative of $\pi_i = \sigma(w^T x_i)$ w.r.t. w

$$\frac{\partial \pi_i}{\partial w} = \sigma(w^T x_i)(1 - \sigma(w^T x_i)) \frac{\partial w^T x_i}{\partial w} = \pi_i (1-\pi_i) x_i. \tag{S.332}$$

Substituting into (S.331),

$$\nabla E(w) = -\sum_{i=1}^{n}\left[y_i\frac{1}{\pi_i}\pi_i(1-\pi_i)x_i + (1-y_i)\frac{-1}{(1-\pi_i)}\pi_i(1-\pi_i)x_i\right] \tag{S.333}$$

$$= -\sum_{i=1}^{n}\left[y_i(1-\pi_i)x_i - (1-y_i)\pi_i x_i\right] = -\sum_{i=1}^{n}\left[y_i x_i - y_i\pi_i x_i - \pi_i x_i + y_i\pi_i x_i\right] \tag{S.334}$$

$$= -\sum_{i=1}^{n}(y_i-\pi_i)x_i = \sum_{i=1}^{n}(\pi_i - y_i)x_i = X(\pi - y). \tag{S.335}$$

The gradient of $w^T\Gamma w$ is Γw. Hence,

$$\nabla\hat{E}(w) = X(\pi - y) + \Gamma w. \tag{S.336}$$

(c) For the Hessian of $\hat{E}(w)$,

$$\nabla^2\hat{E}(w) = \frac{\partial}{\partial w}\frac{\partial}{\partial w^T}\hat{E}(w) = \frac{\partial}{\partial w}\left[(\pi - y)^T X^T + w^T\Gamma\right] \tag{S.337}$$

$$= \frac{\partial}{\partial w}\left[\pi^T X^T - y^T X^T + w^T\Gamma\right]. \tag{S.338}$$

Next, we note that

$$\frac{\partial}{\partial w}w^T = \begin{bmatrix}\frac{\partial}{\partial w_1}\\ \vdots \\ \frac{\partial}{\partial w_d}\end{bmatrix}\left[w_1, \cdots, w_d\right] = I, \tag{S.339}$$

$$\frac{\partial}{\partial w}\pi^T = \frac{\partial}{\partial w}\left[\pi_1, \cdots, \pi_n\right] = \left[\pi_1(1-\pi_1)x_1, \cdots, \pi_n(1-\pi_n)x_n\right] = XR, \tag{S.340}$$

where $R = \text{diag}(\pi_1(1-\pi_1), \cdots, \pi_n(1-\pi_n))$. Finally, substituting into (S.338),

$$\nabla^2\hat{E}(w) = XRX^T + \Gamma. \tag{S.341}$$

(d) The Newton-Raphson iteration is

$$w^{(new)} = w^{(old)} - [\nabla^2 E(w)]^{-1}\nabla E(w) \tag{S.342}$$

$$= w^{(old)} - (XRX^T + \Gamma)^{-1}(X(\pi - y) + \Gamma w^{(old)}) \tag{S.343}$$

$$= (XRX^T + \Gamma)^{-1}(XRX^T + \Gamma)w^{(old)} - (XRX^T + \Gamma)^{-1}(X(\pi - y) + \Gamma w^{(old)}) \tag{S.344}$$

$$= (XRX^T + \Gamma)^{-1}(XRX^T w^{(old)} - X(\pi - y)) \tag{S.345}$$

$$= (XRX^T + \Gamma)^{-1}XR\underbrace{(X^T w^{(old)} - R^{-1}(\pi - y))}_{z} \tag{S.346}$$

$$= (XRX^T + \Gamma)^{-1}XRz \tag{S.347}$$

(e) Γ is added to the term XRX^T which is then inverted. Hence, Γ helps to ensure that we avoid inverting a singular matrix.

(f) Substituting into the objective function $\hat{E}(w)$,

$$\hat{E}_1(w) = E(w) + \frac{1}{2} w^T \Gamma_1 w = E(w) + \frac{1}{2}(\lambda \|\tilde{w}\| + \tilde{b}^2), \tag{S.348}$$

$$\hat{E}_2(w) = E(w) + \frac{1}{2} w^T \Gamma_2 w = E(w) + \frac{1}{2}\lambda \|\tilde{w}\| . \tag{S.349}$$

Hence, the prior covariance Γ_1 applies a squared penalty on $\tilde{b}$, thus favoring smaller $\tilde{b}$. In contrast, prior Γ_2 does not apply a penalty on $\tilde{b}$. In general, we should prefer Γ_2. The value of $\tilde{b}$ depends on the location of the data in the space. Data that is far from the origin will typically have large values of $\tilde{b}$ in order to translate the separating hyperplane into the region where the data lies. It seems unreasonable to penalize such translations.

.....

Problem 9.4 Soft-margin SVM (1-norm penalty)

(a) First, rewrite the soft-margin inequality constraints

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad \Rightarrow \quad y_i(w^T x_i + b) - 1 + \xi_i \geq 0. \tag{S.350}$$

Introducing Lagrange multipliers $\alpha_i \geq 0$ for the soft-margin inequality constraints, and $r_i \geq 0$ for the non-negative slack constraint ($\xi_i \geq 0$), we can obtain the Lagrangian

$$L(w, b, \xi, \alpha, r) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + b) - 1 + \xi_i) - \sum_{i=1}^{n} r_i \xi_i. \tag{S.351}$$

(b) Taking the derivative w.r.t. $\{w, b, \xi\}$ and setting to zero yields the conditions for a minimum,

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \quad \Rightarrow \quad w^* = \sum_{i=1}^{n} \alpha_i y_i x_i. \tag{S.352}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{S.353}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - r_i = 0 \quad \Rightarrow \quad r_i = C - \alpha_i. \tag{S.354}$$

(c) Substituting (S.352) into L,

$$L = \frac{1}{2}\left\|\sum_{i=1}^{n} \alpha_i y_i x_i\right\|^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left(y_i \left[(\sum_{j=1}^{n} \alpha_j y_j x_j)^T x_i + b \right] - 1 + \xi_i \right) - \sum_{i=1}^{n} r_i \xi_i. \tag{S.355}$$

Next, expanding the norm term and collecting terms together,

$$L = \underbrace{\frac{1}{2}(\sum_{i=1}^{n} \alpha_i y_i x_i)^T (\sum_{j=1}^{n} \alpha_j y_j x_j) - \sum_{i=1}^{n} \alpha_i y_i \sum_{j=1}^{n} \alpha_j y_j x_j^T x_i}_{-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_j^T x_i} - \underbrace{\sum_{i=1}^{n} \alpha_i y_i}_{0 \text{ (S.353)}} b + \sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} (\underbrace{C - \alpha_i - r_i}_{0 \text{ (S.354)}}) \xi_i. \tag{S.356}$$

Hence, the dual function is

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j. \tag{S.357}$$

(d) From (S.354), we have $\alpha_i + r_i = C$. Since $r_i \geq 0$ and $\alpha_i \geq 0$, then the upper-bound of α_i is C when $r_i = C$. Hence, we have the equivalent constraint $0 \leq \alpha_i \leq C$. Finally, maximizing the dual function subject to this constraint and (S.353) yields the SVM dual problem,

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \tag{S.358}$$

$$\text{s.t. } \sum_{i=1}^{n} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \forall i. \tag{S.359}$$

(e) For the optimization problem and corresponding Lagrangian,

$$\min_x f(x) \;\; \text{s.t.} \;\; g(x) \geq 0, \quad L(x, \lambda) = f(x) - \lambda g(x), \tag{S.360}$$

the following KKT conditions are satisfied at a solution,

$$\begin{cases} g(x) \geq 0, \\ \lambda \geq 0, \\ \lambda g(x) = 0. \end{cases} \tag{S.361}$$

The last condition implies that either:

- $g(x) = 0$ and $\lambda > 0$, i.e., the equality constraint is *active*.
- $g(x) > 0$ and $\lambda = 0$, i.e., the equality constraint is *inactive*.

Since the SVM problem has several Lagrange multipliers (α_i, r_i), we consider different combinations active/inactive constraints:

LM	slack constraint	LM	margin constraint	interpretation of point x_i
$r_i > 0$	active, $\xi_i = 0$.	$\alpha_i = 0$	inactive, $y_i(w^T x_i + b) > 1$.	The point is correctly classified and beyond the margin.
$r_i > 0$	active, $\xi_i = 0$.	$\alpha_i > 0$	active, $y_i(w^T x_i + b) = 1$.	The point is correctly classified and on the margin.
$r_i = 0$	inactive $\xi_i > 0$	$\alpha = C$, using (S.354).	active, $y_i(w^T x_i + b) = 1 - \xi_i$.	Since $\xi_i > 0$, then the point is violating the margin (i.e., inside the margin).

.....

Problem 10.12 Gaussian process regression - nonlinear Bayesian regression

(a) Substituting $\hat{\mu}_\theta$ into the predictive mean $\hat{\mu}_*$,

$$\hat{\mu}_* = \phi_*^T \hat{\mu}_\theta = \phi_*^T \underbrace{(\Gamma^{-1} + \Phi \Sigma^{-1} \Phi^T)^{-1} \Phi \Sigma^{-1}}_{(P^{-1} + B^T R^{-1} B)^{-1} B^T R^{-1} = P B^T (B P B^T + R)^{-1}} y \tag{S.362}$$

$$= \phi_*^T \Gamma \Phi (\Phi^T \Gamma \Phi + \Sigma)^{-1} y \tag{S.363}$$

$$= \phi_*^T \Gamma \Phi (\Phi^T \Gamma \Phi + \sigma^2 I)^{-1} y, \tag{S.364}$$

where we use the matrix inverse identity (1.33) in Problem 1.15.

(b) Substituting the covariance $\hat{\Sigma}_\theta$ into the predictive variance $\hat{\sigma}_*^2$ gives

$$\hat{\sigma}_*^2 = \phi_*^T \hat{\Sigma}_\theta \phi_* = \phi_*^T \underbrace{(\Gamma^{-1} + \Phi\Sigma^{-1}\Phi^T)^{-1}}_{(A^{-1}+UC^{-1}V^T)^{-1} = A - AU(C+V^TAU)^{-1}V^TA)} \phi_* \tag{S.365}$$

$$= \phi_*^T(\Gamma - \Gamma\Phi(\Sigma + \Phi^T\Gamma\Phi)^{-1}\Phi^T\Gamma)\phi_* \tag{S.366}$$

$$= \phi_*^T\Gamma\phi_* - \phi_*^T\Gamma\Phi(\Phi^T\Gamma\Phi + \sigma^2 I)^{-1}\Phi^T\Gamma\phi_*, \tag{S.367}$$

where we have used the matrix inverse identity (1.36) in Problem 1.15.

(c) Using the matrix inverse identities, the expressions for $\hat{\mu}_*$ and $\hat{\sigma}_*^2$ now depend on the data through inner products $\Phi^T\Gamma\Phi$. Hence, we define the kernel function as $k(x_i, x_j) = \phi(x_i)^T\Gamma\phi(x_j)$, resulting in

$$\hat{\mu}_* = k_*^T(K + \sigma^2 I)^{-1} y \tag{S.368}$$

$$\hat{\sigma}_*^2 = k_{**} - k_*^T(K + \sigma^2 I)^{-1} k_*, \tag{S.369}$$

where $K = \begin{bmatrix} k(x_1,x_1) & \cdots & k(x_1,x_n) \\ \vdots & \ddots & \vdots \\ k(x_n,x_1) & \cdots & k(x_n,x_n) \end{bmatrix}$ is the kernel matrix with entries $k(x_i, x_j)$, $k_* = \begin{bmatrix} k(x_*,x_1) \\ \vdots \\ k(x_*,x_n) \end{bmatrix}$ is the test kernel vector, and $k_{**} = k(x_*, x_*)$. The prior covariance Γ is embedded into the kernel function – one interpretation is that the prior covariance parameter is changed into the parameters of the kernel function.

(d) Note that $z = (K + \sigma^2 I)^{-1} y$ is fixed for a given training set. We have

$$\hat{\mu}_* = k_*^T z = \sum_{i=1}^n z_i k(x_*, x_i). \tag{S.370}$$

In other words, the z contains coefficients on the kernel functions centered at each x_i. Substituting for different kernel functions, we can get the forms of the posterior mean $\hat{\mu}_*$ as a function of x_*,

- linear: $k(x_i, x_j) = x_i^T x_j$. The posterior mean is

$$\hat{\mu}_* = \sum_{i=1}^n z_i (x_*^T x_i) = x_*^T \left(\sum_{i=1}^n z_i x_i \right), \tag{S.371}$$

  which is a linear function in x_*.

- polynomial: $k(x_i, x_j) = (x_i^T x_j + 1)^2$. The posterior mean is

$$\hat{\mu}_* = \sum_{i=1}^n z_i (x_*^T x_i + 1)^2 = \sum_{i=1}^n z_i ((x_*^T x_i)^2 + 2x_*^T x_i + 1) \tag{S.372}$$

$$= \sum_{i=1}^n z_i (x_*^T x_i x_i^T x_* + 2x_*^T x_i + 1) = x_*^T \left(\sum_{i=1}^n z_i x_i x_i^T \right) x_* + 2x_*^T \left(\sum_{i=1}^n z_i x_i \right) + \sum_{i=1}^n z_i, \tag{S.373}$$

  which is a quadratic function in x_*.

- RBF: $k(x_i, x_j) = e^{-\alpha \|x_i - x_j\|^2}$. The posterior mean is

$$\hat{\mu}_* = \sum_{i=1}^{n} z_i e^{-\alpha \|x_* - x_i\|^2}. \tag{S.374}$$

  This looks a like a kernel density estimate, except we don't have any constraints on the weights z_i, so the function can be negative.

- periodic: $k(x_i, x_j) = e^{-\sin^2 \frac{(x_i - x_j)}{2}}$. The posterior mean is

$$\hat{\mu}_* = \sum_{i=1}^{n} z_i e^{-\sin^2 \frac{(x_* - x_i)}{2}} = \sum_{i=1}^{n} z_i \sum_{m=0}^{\infty} \frac{1}{m!} \left(-\sin^2 \frac{(x_* - x_i)}{2} \right)^m \tag{S.375}$$

$$= \sum_{i=1}^{n} z_i + \sum_{m=1}^{\infty} \frac{(-1)^m}{m!} \sum_{i=1}^{n} z_i \sin^{2m} \frac{(x_* - x_i)}{2}, \tag{S.376}$$

  where we have used the Taylor expansion of the exponential function $e^x = \sum_{m=0}^{\infty} \frac{x^m}{m!}$. Note that $\sin^{2m}(\theta)$ can be rewritten as a linear combination of cosines with frequencies that are integer multiples of the base frequency,

$$\sin^{2m}(\theta) = a_{m,0} + \sum_{j=1}^{m} a_{m,j} \cos(2j\theta), \tag{S.377}$$

  for some coefficients $\{a_{m,j}\}$ (the actual formula is a little messy). Substituting into (S.376),

$$\hat{\mu}_* = \sum_{i=1}^{n} z_i + \sum_{m=1}^{\infty} \frac{(-1)^m}{m!} \sum_{i=1}^{n} z_i \left[a_m + \sum_{j=1}^{m} a_{m,j} \cos(j(x_* - x_i)) \right] \tag{S.378}$$

$$= b_0 + \sum_{m=1}^{\infty} b_m \sum_{i=1}^{n} z_i \cos(m(x_* - x_i)), \tag{S.379}$$

  for some weights $\{b_m\}$. Note that a linear combination of cosines of the same frequency (m) but with different phase shifts x_i is also a scaled cosine of frequency m. Hence, the mean function finally reduces to

$$\hat{\mu}_* = w_0 + \sum_{m=1}^{\infty} w_m \cos(m(x_* - c_m)), \tag{S.380}$$

  for some weights $\{w_m\}$ and phase offsets $\{c_m\}$. In other words, the mean function is a weighted sum of cosines, each with a frequency that is a multiple of the base frequency of 1. Hence, the mean function is a periodic function with frequency 1. Here we have assumed that the frequency is 1, but we could easily add a parameter θ to change the base frequency, as in $\sin^2(\theta(x_i - x_j)/2)$.

  Figure 12 shows examples of the above four kernels.

(e) Since $(K + \sigma^2 I)$ is a positive definite matrix, as is its inverse, then the smallest value that $k_*^T (K + \sigma^2 I)^{-1} k_*$ can take is 0. Hence the predictive variance is bounded by

$$0 \leq \hat{\sigma}_*^2 \leq k_{**}. \tag{S.381}$$

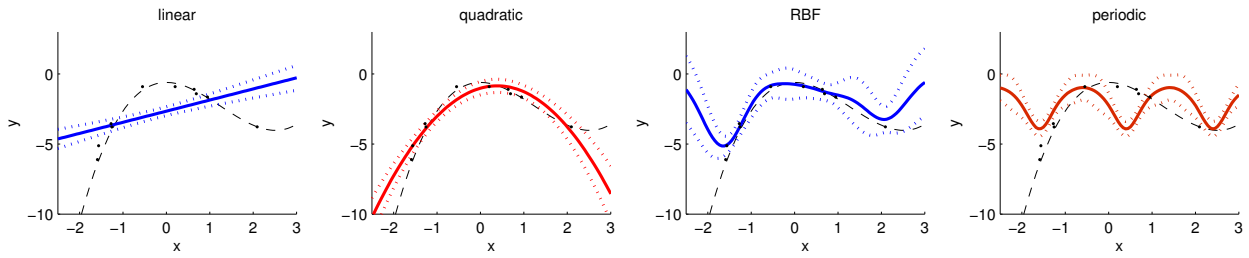

Figure 12: GP functions learned using linear, quadratic, RBF, and periodic kernels. The dashed line is the ground-truth function. The thick line is the posterior mean, while the dotted lines show 2 standard-deviations around the mean.

The predictive variance is largest when

$$k_*^T(K + \sigma^2 I)^{-1} k_* = 0 \quad \Rightarrow \hat{\sigma}_*^2 = k_{**}. \tag{S.382}$$

This occurs when the vector $k_* = 0$, i.e., $k(x_*, x_i) = 0$ for all training samples x_i. This means that x_* is different than all examples in the training set. Formally, $\langle \phi(x_*), \phi(x_i) \rangle$=0, and hence x_* is orthogonal to all x_i in the high-dimensional feature space.

The predictive variance is smallest when

$$k_{**} = k_*^T(K + \sigma^2 I)^{-1} k_* \quad \Rightarrow \hat{\sigma}_*^2 = 0. \tag{S.383}$$

Consider the case when the test point x_* is the same as the first point x_1, and different from all other points x_i, $i > 2$. Also assume the extreme case where $k(x_*, x_1) = k(x_*, x_*) = k_{**} = \alpha$ and $k(x_*, x_i) = 0$ for $i > 2$. E.g., this could occur when using a Gaussian kernel with the bandwidth is set very small. The test kernel vector is

$$k_* = \begin{bmatrix} k(x_*, x_1) \\ k(x_*, x_2) \\ \vdots \\ k(x_*, x_n) \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{S.384}$$

Also, the kernel matrix can be decomposed in blocks as

$$K = \left[\begin{array}{c|c} k(x_1, x_1) & \cdots \quad k(x_1, x_n) \\ \hline \vdots & \\ k(x_n, x_1) & \qquad K_2 \end{array}\right] = \begin{bmatrix} \alpha & 0^T \\ 0 & A \end{bmatrix}, \tag{S.385}$$

since $x_1 = x_*$. Then we have

$$(K + \sigma^2 I)^{-1} = \begin{bmatrix} \alpha + \sigma^2 & 0^T \\ 0 & A + \sigma^2 I \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{\alpha + \sigma^2} & 0^T \\ 0 & (A + \sigma^2 I)^{-1} \end{bmatrix}, \tag{S.386}$$

which uses the property that the inverse of a block-diagonal matrix is a block-diagonal matrix of the inverses. Substituting into the quadratic term,

$$k_*^T(K + \sigma^2 I)^{-1} k_* = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}^T \begin{bmatrix} \frac{1}{\alpha + \sigma^2} & 0^T \\ 0 & (A + \sigma^2 I)^{-1} \end{bmatrix} \begin{bmatrix} \alpha \\ 0 \end{bmatrix} = \frac{\alpha^2}{\alpha + \sigma^2}. \tag{S.387}$$

Hence, when $\sigma^2 = 0$, then

$$k_*^T (K + \sigma^2 I)^{-1} k_* = \alpha = k_{**} \quad \Rightarrow \quad \hat{\sigma}_*^2 = 0. \tag{S.388}$$

In other words, the predictive variance will be 0 when the observation noise is 0, and the test point is the same as one training point and dissimilar to all other training points. This is the extreme case. In general, the predictive variance will be small when the observation noise is small and the test point is very similar to only one training point and very dissimilar to all others, according to the kernel function (i.e., the kernel function is very localized, or has small bandwidth).

. . . . . .