An introduction to a typical classification problem, both binary and multi-class, is provided in Appendix A. It covers the material from the first portfolio, and the rest of this report will build on those concepts.

1 Frequentist Classification

1.1 Least Squares Classification

Returning to binary classification, we can try to fit an LS estimate on D

$$\mathbf{w}_{LS} \coloneqq \arg\min_{\mathbf{w}} \sum_{i \in D} [y_i - f(\mathbf{x}_i; \mathbf{w})]^2$$
$$\text{and take } \hat{y} = \text{sign}\left(f(\mathbf{x}_i; \mathbf{w}_{LS})\right)$$

Remark 1.1 *We can also use a feature transform $\boldsymbol{\phi}$ in cases were our data is not separable in the original space but is in the feature space created by $\boldsymbol{\phi}$ (e.g. classes that form an approximate circle around each other).*

In the multi-class example, we can replace $y_i = k$ with a vector $\mathbf{t}_i$ such that the i^{th} element is 1 while all other entries are 0 (this is called one-hot encoding). Then

$$\mathbf{W}_{LS} = \arg\min_{\mathbf{W}} \sum_{i \in D} ||\mathbf{t}_i - \mathbf{f}(\mathbf{x}; \mathbf{W})||^2$$
$$\text{where } \mathbf{W} \in \mathbb{R}^{(d+1)\times K},\ \tilde{\mathbf{x}}_i = [\mathbf{x}_i^T, 1]^T \text{ and } \mathbf{f}(\mathbf{x}; \mathbf{W}) = \mathbf{W}^T \tilde{\mathbf{x}}$$

and we predict

$$\hat{k} = \arg\max_k f^{(k)}(\mathbf{x}; \mathbf{W}) = \arg\max_k \left(\mathbf{w}_{LS}^{(k)}\right)^T \tilde{\mathbf{x}}$$

However, we can see that there are a few drawbacks to this method:

- The shape of the square loss function makes it so that correctly classified data points far away from decision boundary will still have a lot of influence over the decision boundary
- It lacks a probabilistic interpretation - it does not correspond to the maximum likelihood solution.

1.2 Fisher Discriminant Analysis

With this method, we make use of the inner product $\langle \mathbf{w}; \mathbf{x} \rangle$, where, geometrically speaking, $\mathbf{x}$ is embedded onto a 1-dimensional line along the direction of $\mathbf{w}$. We would then generally want to choose a $\mathbf{w}$ that separates the classes well, where points in the same class are close to each other and points in different classes are far apart. More formally:

- The within-class scatterness $\left(\text{given by } s_{\mathbf{w},k} = \sum_{i,y_i=k} (\mathbf{w}^T\mathbf{x}_i - \hat{\mu}_k)^2, \text{ where } \hat{\mu}_k = \frac{1}{n_k} \sum_{i,y_i=k} \mathbf{w}^T\mathbf{x}_i\right)$ is small.
- The between-class scatterness $\left(\text{given by } s_{b,k} = n_k(\hat{\mu}_k - \hat{\mu})^2, \text{ where } \hat{\mu} = \frac{1}{n} \sum_{i \in D} \mathbf{w}^T\mathbf{x}_i\right)$ is large.

Hence, we choose $\mathbf{w}$ that satisfies:

$$\arg\max_{\mathbf{w}} \frac{\sum_k s_{b,k}}{\sum_k s_{\mathbf{w},k}}$$

However, this method doesn't give us a prediction function f. Further, the sign of $f(\mathbf{x}; \mathbf{w}_{FDA})$ in the binary case does not indicate the prediction, and there is no way to control the prediction accuracy.

2 Probabilistic Classification

In this approach, we minimise the expected loss:

$$\hat{y} \coloneqq \arg\min_{y_0} \mathbb{E}_{p(y|\mathbf{x})}[L(y, y_0)|xv]$$

For this we need to obtain $p(y|\mathbf{x})$, which we can do either generatively or discriminatively.

2.1 Generative Classifiers

In this case we infer $p(y|\mathbf{x})$ from $p(\mathbf{x}|y)$ and $p(y)$, hence we start with a model for $p(\mathbf{x}|y)$.

2.1.1 Continuous Input Variables

If $\mathbf{x}$ is continuous, we can choose a multivariate normal:

$$p(\mathbf{x}|y = k; \mathbf{w}) = \mathcal{N}_{\mathbf{x}}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

If we assume the data is IID and all classes have the same covariance $\boldsymbol{\Sigma}$, the likelihood is

$$p(D|\mathbf{w}) = \prod_{i \in D} p(\mathbf{x}_i, y_i|\mathbf{w}) = \prod_{i \in D} p(\mathbf{x}_i|y_i; \mathbf{w})p(y_i) = \prod_{i \in D} \mathcal{N}_{\mathbf{x}_i}(\boldsymbol{\mu}_{y_i}; \boldsymbol{\Sigma})p(y_i)$$

and so we find the maximum likelihood estimates as

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i \in D} \mathbf{x}_i$$
$$\hat{\boldsymbol{\Sigma}} = \sum_k \frac{n_k}{n} \frac{1}{n_k} \sum_{i \in D, y_i = k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T$$

with the proof given in Appendix B.1. We then predict the response as $\hat{y} = \arg\max_y p(y|\mathbf{x}; \hat{\mathbf{w}}) = \arg\max_y p(\mathbf{x}|y; \hat{\mathbf{w}})p(y)$.

Further, in the multivariate normal model with shared covariance, the decision boundary is piece-wise linear, the proof for which is given in Appendix B.1.

2.1.2 Discrete Input Variables

When $\mathbf{x}$ is discrete (for example representing frequencies of words in an email) then we might take a "naive Bayes" approach, and assume each $x^{(i)}$ follows a multinomial distribution, and so

$$p(\mathbf{x}|y = k) \propto \prod_{i=1,\ldots,d} \beta(i|y = k)^{x^i}$$

where $\beta(i|y = k)$ is the probability the word i occurs in a email of class k, which we estimate as

$$\beta(i|y = k) \approx \frac{\sum_{j \in D, y_j = k} x_j^{(i)}}{\sum_{j \in D, y_j = k} \sum_{i=1}^{d} x_j^{(i)}}$$

i.e. the proportion of word i appearing in class k in relation to all emails. This estimate is also consistent with the MLE, the proof for which is provided in Appendix B.1

2.2 Discriminative Classifiers

In this approach we infer $p(y|\mathbf{x})$ directly from our dataset D. We can use Bayes rule to see

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|y)p(y)}{\sum_{y'} p(\mathbf{x}, y')} = \frac{p(\mathbf{x}|y)p(y)}{\sum_{y'} p(\mathbf{x}|y')p(y')}$$

so if $y \in \{-1, +1\}$

$$p(y = 1|\mathbf{x}) = \frac{p(\mathbf{x}|y = 1)p(y = 1)}{p(\mathbf{x}|y' = 1)p(y' = 1) + p(\mathbf{x}|y' = -1)p(y' = -1)}$$

Further if $p(\mathbf{x}|y)p(y) > 0$

$$p(y = 1|\mathbf{x}) = \frac{1}{1 + \frac{p(\mathbf{x}|y=-1)p(y=-1)}{p(\mathbf{x}|y=1)p(y=1)}}$$

and so we have an expression for $p(y|\mathbf{x})$ in terms of the density ratio, which we can model as

$$f(\mathbf{x}; \mathbf{w}) = \log \left[\frac{p(\mathbf{x}|y = 1)p(y = 1)}{p(\mathbf{x}|y = -1)p(y = -1)} \right]$$

where f is linear as usual.

If we create a sigmoid function $\sigma(t) = \frac{1}{1+\exp(-t)}$, we can write $p(y|\mathbf{x}; \mathbf{w}) = \sigma(f(\mathbf{x}; \mathbf{w}) \cdot y)$ for a simpler expression. In the special case where $p(\mathbf{x}|y)$ is a multivariate normal with shared covariance, we can obtain the expression for $\mathbf{w}$ as shown in Appendix B.2.

If we further assume D contains IID data, the likelihood is $p(D; \mathbf{w}) = \prod_{i \in D} p(y_i|\mathbf{x}_i; \mathbf{w})$ and so

$$\mathbf{w}_{MLE} = \arg\max_{\mathbf{w}} \sum_{i \in D} \log(\sigma(f(\mathbf{x}_i; \mathbf{w}) \cdot y_i))$$

This procedure is commonly referred to as logistic regression, even though this is not in fact a regression. The decision functions given by logistic regression will take the form of $f(\mathbf{x}; \mathbf{w}) = p(Y|\mathbf{x}; \mathbf{w}) - c$ where $c \in [0, 1]$

Remark 2.1 *As in the least squares case discussed in Section 1.1, we can use feature transforms $\boldsymbol{\phi}$ to create a linear classifier. Unlike least squares however, this method has the advantage of not begin affected by outliers far away from the decision boundary.*

We are now ready to estimate $p(y|\mathbf{x}; \mathbf{w})$. Given priors on $\mathbf{w}$ we can obtain the MAP estimate

$$\begin{aligned} \mathbf{w}_{MAP} &= \arg\max_{\mathbf{w}} \sum_{i \in D} \log \sigma(f(\mathbf{x}_i; \mathbf{w}) \cdot y_i) p(\mathbf{w})) \\ &= \arg\max_{\mathbf{w}} \sum_{i \in D} \log \sigma(f(\mathbf{x}_i; \mathbf{w}) \cdot y_i) + \log p(\mathbf{w}) \end{aligned}$$

or we can go for the full probabilistic approach

$$p(y|\mathbf{x}) = \int p(y|\mathbf{x}; \mathbf{w}) p(\mathbf{w}|D) d\mathbf{w} \propto \int p(y|\mathbf{x}; \mathbf{w}) p(D|\mathbf{w}) p(\mathbf{w}) d\mathbf{w}$$

however we cannot solve this analytically in the classification case.

Returning to the multi-class case, we can easily extend the logistic regression to get

$$p(y = k|\mathbf{x}) = \frac{p(\mathbf{x}|y = k)p(y = k)}{\sum_{k'} p(\mathbf{x}|y = k')p(y = k')}$$

and we can use one-hot encoding, writing y_i as $\mathbf{t}_i \in \mathbb{R}^K$, to get

$$\mathbf{w}_{MLE} = \arg\max_{\mathbf{w}} \sum_{i \in D} \log \sigma(\mathbf{f}(\mathbf{x}_i; \mathbf{w}), \mathbf{t}_i)$$
$$\text{where } \sigma(\mathbf{f}, \mathbf{t}) = \frac{\exp \langle \mathbf{f}, \mathbf{t} \rangle}{\sum_k \exp f^{(k)}}$$

A probabilistic interpretation of multiclass logistic regression is provided in Appendix B.2.

Again, we cannot obtain a closed-form solution for this and so we need numerical methods.

3 Support Vector Machines

Rather than the previous sections which attempt to find an adequate boundary, support vector machines attempt to identify the best decision boundary. To do this we consider the generalisation principle, which aims to choose the decision boundary which performs the best on unseen data, rather than the observed training data.

We can measure this using the error margin, the width of which is the distance between the boundary and the closest data point to the boundary. A thin margin would mean the boundary is susceptible to misclassification when random perturbations are applied to the data. A thick margin however would mean the classifier is robust.

For the rest of this section we will only consider binary classification, and so we can write the function $f(\mathbf{x}; \mathbf{w}) = \langle \mathbf{w}', \mathbf{x} \rangle + w_0$. Then we can classify any points where $f(\mathbf{x}; \mathbf{w}) > 1$ as $+$ and any points where $f(\mathbf{x}; \mathbf{w}) < 1$ as $-$.

We further find that the distance between the center of this tube and each edge is $\frac{1}{||\mathbf{w}'||}$, this is shown in Appendix C. This means the optimal decision function $f_{\mathbf{w}}$ is given by the widest error margin, which is formalised in the following minimisation problem

$$\text{Minimise } ||\mathbf{w}'||^2$$
$$\text{Subject to } y_i f(\mathbf{x}_i; \mathbf{w}) \geq 1$$

However, this only works when the data is separable, when this is not true, we can adapt the problem.

3.1 Soft-Margin Classifier

We apply a compensation ϵ_i to ensure each point is on the correct side of the boundary and make sure these are as small as possible by including them in the objective.

$$\text{Minimise } ||\mathbf{w}'||^2 + \sum_i \epsilon_i$$
$$\text{Subject to } y_i f(\mathbf{x}_i; \mathbf{w}) + \epsilon_i \geq 1$$
$$\epsilon_i \geq 0$$

Although constrained optimisation problems can be tricky to solve, we can use the Lagrangian dual to transform this into the unconstrained problem

$$\min_{\mathbf{w}} ||\mathbf{w}'||^2 + \begin{cases} 0 & \text{if } y_i f_{\mathbf{w}}(\mathbf{x}_i) \geq 0 \\ 1 - y_i f_{\mathbf{w}}(\mathbf{x}_i) & \text{if } y_i f_{\mathbf{w}}(\mathbf{x}_i) < 0 \end{cases}$$

The proof for this is given in Appendix C.1

A Introduction to Classification

The general set-up of a classification problem has the following components:

- Input of covariates $\mathbf{x} \in \mathbb{R}^d$
- Output $y \in \{1, \ldots, K\}$
- A rule or set of rules to predict the classification y based on the covariates $\mathbf{x}$

A.1 Binary Classification

In the case when there are only two options for y (-1 or $+1$), the classification rule takes the form of a decision boundary $f(\mathbf{x}) = 0$, with $f(\mathbf{x}) < 0$ indicating $y = -1$ and $f(\mathbf{x}) > 0$ indicating $y = +1$. Further, it is often assumed f is a linear function of the form

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + w_0$$

A.2 Multi-class Classification

Here we cannot define the prediction of y based on the sign of a single function f. There are a couple modifications of this approach we can try though.

One v. The Other We construct $K - 1$ classifiers f_i, where for the i^{th} function, we consider the i^th class against all the other classes. However the issue with this is that there will always be at least one region which is not associated with any of the K classes.

One v. One Here we construct a classifier for each pairwise combination of classes, and take the mode of all the predictions. However this will also create regions for which there is not a clear consensus.

Moving away from using the sign of f to make a prediction, we could construct a function $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^K$ which gives a value in the real numbers for each class K. Then we take the prediction given $\mathbf{x}$ to be

$$\hat{k} = \arg\max_k f^{(k)}(\mathbf{x})$$

B Bayesian Classification

B.1 Generative Classification

Proof of Piece-wise Linear Decision Boundary:

We know the boundary is given by the set

$$\{\mathbf{x} | p(y = k|\mathbf{x}; \hat{\mathbf{w}}) = p(y = k'|\mathbf{x}; \hat{\mathbf{w}}),\ \forall k \neq k'\} = \left\{ \mathbf{x} \middle| \frac{p(\mathbf{x}|y = k; \hat{\mathbf{w}})p(y = k)}{p(\mathbf{x}|y = k'; \hat{\mathbf{w}})p(y = k')} = 1,\ \forall k \neq k' \right\}$$

Taking the log of both sides ofthis inner relation we get

$$\log\left(\frac{p(\mathbf{x}|y = k; \hat{\mathbf{w}})}{p(\mathbf{x}|y = k'; \hat{\mathbf{w}})}\right) + \log\left(\frac{p(y = k)}{p(y = k')}\right) = 0$$

Expanding the first term we see

$$\begin{aligned}
\log\left(\frac{p(\mathbf{x}|y=k;\hat{\mathbf{w}})}{p(\mathbf{x}|y=k';\hat{\mathbf{w}})}\right) &= \log\left(\frac{(2\pi)^{-\frac{n}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right)}{(2\pi)^{-\frac{n}{2}}|\boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{k'})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{k'})\right)}\right) \\
&= -\frac{1}{2}\left((\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k) - (\mathbf{x}-\boldsymbol{\mu}_{-k'})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{k'})\right) \\
&= -\frac{1}{2}\left(\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - 2\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\mathbf{x} + \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k - \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} + 2\boldsymbol{\mu}_{k'}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \boldsymbol{\mu}_{k'}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{k'}\right) \\
&= -\frac{1}{2}\left((\boldsymbol{\mu}_k - \boldsymbol{\mu}_{\prime k})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_{k'}) - 2(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})\boldsymbol{\Sigma}^{-1}\mathbf{x}\right)
\end{aligned}$$

and so the inner relation is

$$(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{\prime k})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_{k'}) + \log\left(\frac{p(y=k)}{p(y=k')}\right) = 0$$

which is a linear function of $\mathbf{x}$. Since the boundary is the union of all these linear components, we conclude the it is piece-wise linear.

Proof of ML estimates for MVN input:

The maximum likelihood estimates must satisfy

$$\hat{\boldsymbol{\mu}}_{1,\ldots,K}, \hat{\boldsymbol{\Sigma}} = \underset{\boldsymbol{\mu}_{1,\ldots,K},\boldsymbol{\Sigma}}{\arg\max} \sum_{i\in D} \log\left[\mathcal{N}_{\mathbf{x}_i}(\boldsymbol{\mu}_{y_i};\boldsymbol{\Sigma})p(y_i)\right]$$

We can ignore the $p(y_i)$ terms as they are constants not involving the parameters we want to estimate. Thus we can write the objective function for this problem as

$$S(\boldsymbol{\mu}_{1,\ldots,K},\boldsymbol{\Sigma}) = \sum_{i\in D}\left[-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_{y_i})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_{y_i}) + \frac{n}{2}|\boldsymbol{\Sigma}^{-1}|\right]$$

Then we can first differentiate with respect to $\boldsymbol{\mu}_k$ and set the result to 0:

$$\begin{aligned}
\frac{\partial S}{\partial \boldsymbol{\mu}_k} &= \sum_{i\in D, y_i=k} -(\mathbf{x}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}^{-1} \\
\implies n_k\hat{\boldsymbol{\mu}}_k = \textstyle\sum_{i\in D,y_i=k}\mathbf{x}_i \implies \hat{\boldsymbol{\mu}}_k &= \frac{1}{n_k}\sum_{i\in D,y_i=k}\mathbf{x}_i
\end{aligned}$$

Then we differentiate with respect to $\boldsymbol{\Sigma}$:

$$\begin{aligned}
\frac{\partial S}{\partial \boldsymbol{\Sigma}} &= \frac{n}{2}\boldsymbol{\Sigma}^T - \frac{1}{2}\sum_{i\in D}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \\
\implies n\hat{\boldsymbol{\Sigma}} - \sum_{i\in D}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T &= \mathbf{0} \\
\implies \hat{\boldsymbol{\Sigma}} = \sum_{k=1}^{K}\frac{n_k}{n}\frac{1}{n_k}\sum_{i\in D,y_i=k}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T
\end{aligned}$$

as desired.

Proof of ML Multinomial Parameters

If we assume $\mathbf{x} \sim \text{Multinomial}(\mathbf{p})$, we get

$$p(\mathbf{x}|\mathbf{p}, n) = \binom{n!}{x^{(1)}! \dots x^{(d)}} \prod_{i=1,\dots,d} p_i^{x^{(i)}} = n! \prod_{i=1,\dots,d} \frac{p_i^{x^{(i)}}}{x^{(i)}!}$$

where n is the total number of occurrences. Assuming a dataset D containing multinomial N IID observations, we get the likelihood

$$p(D|\mathbf{p}, n) = \prod_{j \in D} n_j! \prod_{i=1,\dots,d} \frac{p_i^{x_j^{(i)}}}{x_j^{(i)}!}$$

Taking the log we get

$$\ell(D|\mathbf{p}) = \log\left(\sum_{j \in D} n_j!\right) + \sum_{j \in D}\left[\sum_{i=1,\dots,d}\left[x_j^{(i)} \log(p_i) - \log(x_j^{(i)}!)\right]\right]$$

Before solving for the maximum likelihood estimate, we need to consider the constraint $\sum_{i=1,\dots,d} p_i = 1$. The Lagrangian for this problem is

$$\mathcal{L}(\mathbf{p}, \lambda) = \ell(D|\mathbf{p}) + \lambda\left(1 - \sum_{i=1,\dots,d} p_i\right)$$

Now differentiating with respect to p_i we see

$$\begin{aligned}\frac{\partial}{\partial p_i}\mathcal{L}(\mathbf{p}, \lambda) &= \frac{\partial}{\partial p_i}\left(\log\left(\sum_{j \in D} n_j!\right) + \sum_{j \in D}\left[\sum_{i=1,\dots,d}\left[x_j^{(i)} \log(p_i) - \log(x_j^{(i)}!)\right]\right] + \lambda\left(1 - \sum_{i=1,\dots,d} p_i\right)\right) \\ &= \frac{\sum_{j \in D} x_j^{(i)}}{p_i} - \lambda\end{aligned}$$

and so we get

$$\hat{p}_i = \frac{\sum_{j \in D} x_j^{(i)}}{\lambda}$$

To get λ we use the initial constraint

$$1 = \sum_{i=1,\dots,d} \hat{p}_i = \sum_{i=1,\dots,d} \frac{\sum_{j \in D} x_j^{(i)}}{\lambda} = \frac{\sum_{j \in D} n_j}{\lambda} \implies \lambda = \frac{1}{\sum_{j \in D} n_j}$$

hence

$$\hat{p}_i = \frac{\sum_{j \in D} x_j^{(i)}}{\sum_{j \in D} n_j}$$

Naive Bayes Classifier under ML Framework:

Again, assuming each $x^{(i)}$ follows a multinomial distribution, we see

$$p(\mathbf{x}|y = k) \propto \prod_{i=1,\dots,d} \beta(i|y = k)^{x^i}$$

Here $\beta(i|y=k)$ is simply a parameter in a multinomial distribution for data within class k, hence we only consider the frequencies in $(\mathbf{x}_j, y_j) \in D$ such that $y_j = k$. Thus using the expression for the MLE of a multinomial distribution to obtain:

$$\hat{\beta}(i|y=k) = \frac{\sum_{j\in D, y_j=k} x_j^{(i)}}{\sum_{j\in D, y_j=k} n_j} = \frac{\sum_{j\in D, y_j=k} x_j^{(i)}}{\sum_{j\in D, y_j=k} \sum_{i'=1,\ldots,d} x_j^{(i')}}$$

B.2 Discriminative Classifiers

Prediction in the case of multivariate normal $p(\mathbf{x}|y)$

For a shared $\mathbf{\Sigma}$, we write

$$p(\mathbf{x}|y=1) = \mathcal{N}(\boldsymbol{\mu}_+, \mathbf{\Sigma}) \qquad\qquad p(\mathbf{x}|y=-1) = \mathcal{N}(\boldsymbol{\mu}_-, \mathbf{\Sigma})$$

Then

$$\begin{aligned} f(\mathbf{x};\mathbf{w}) &= \log\left[\frac{p(\mathbf{x}|y=1)p(y=1)}{p(\mathbf{x}|y=-1)p(y=-1)}\right] \\ &= \log\left[\frac{p(\mathbf{x}|y=1)}{p(\mathbf{x}|y=-1)}\right] + \log\left[\frac{p(y=1)}{p(y=-1)}\right] \end{aligned}$$

where

$$\begin{aligned} \log\left[\frac{p(\mathbf{x}|y=1)}{p(\mathbf{x}|y=-1)}\right] &= \log\left[\frac{(2\pi)^{-n/2}|\mathbf{\Sigma}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_+)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_+)\right)}{(2\pi)^{-n/2}|\mathbf{\Sigma}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_-)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_-)\right)}\right] \\ &= -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_+)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_+) + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_-)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_-) \\ &= \frac{1}{2}\left(\mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x} - 2\mathbf{x}^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_- + \boldsymbol{\mu}_-^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_- - \mathbf{x}^T\mathbf{\Sigma}^{-1}\mathbf{x} + 2\mathbf{x}^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_+ - \boldsymbol{\mu}_+^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_+\right) \\ &= \frac{1}{2}\left((\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)^T\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_- + \boldsymbol{\mu}_+) + 2\mathbf{x}^T\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)\right) \\ &= \left\langle \mathbf{x}, \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)\right\rangle + \frac{1}{2}(\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)^T\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_- + \boldsymbol{\mu}_+) \end{aligned}$$

and so

$$f(\mathbf{x};\mathbf{w}) = \left\langle \mathbf{x}, \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)\right\rangle + \frac{1}{2}(\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)^T\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_- + \boldsymbol{\mu}_+) + \log\left[\frac{p(y=1)}{p(y=-1)}\right]$$

Thus there is $\mathbf{w}^* = [\mathbf{w}'^{*T}, w_0^*]^T$ such that $p(y|\mathbf{x}) = \sigma\left(\left(\left\langle \mathbf{x}; \mathbf{w}'^*\right\rangle + w_0^*\right)\cdot y\right)$ given by

$$\mathbf{w}'^* = \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-) \qquad w_0^* = \frac{1}{2}(\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)^T\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_- + \boldsymbol{\mu}_+) + \log\left[\frac{p(y=1)}{p(y=-1)}\right]$$

Probabilistic Multi-class Logistic Regression

In the multi-class logistic regression, we have

$$p(y=k|\mathbf{x}) = \frac{p(\mathbf{x}|y=k)p(y=k)}{\sum_{k'} p(\mathbf{x}|y=k')p(y=k')}$$

As in the binary case, provided $p(\mathbf{x}|y=k)p(y=k) = 0$,

$$p(y=k|\mathbf{x}) = \frac{1}{1 + \frac{\sum_{k'\neq k} p(\mathbf{x}|y=k')p(y=k')}{p(\mathbf{x}|y=k)p(y=k)}}$$

and we can model the log of the density ratio for each k using a linear function $f^{(k)}$

$$f^{(k)}(\mathbf{x}, \mathbf{w}_k) = \log \left[\frac{p(\mathbf{x}|y=k)p(y=k)}{\sum_{k' \neq k} p(\mathbf{x}|y=k')p(y=k')} \right]$$

Then using the sigmoid function again,

$$p(y=k|\mathbf{x}; \mathbf{w}_k) = \sigma\left(f^{(k)}(\mathbf{x}; \mathbf{w}_k)\right)$$

In the one-hot encoding case, we see

$$p(\mathbf{t}|\mathbf{x}; \mathbf{W}) = \sigma(\mathbf{f}(\mathbf{x}, \mathbf{W}), \mathbf{t})$$

where the decision function becomes a vector of log density ratios $\mathbf{f} = [f^{(1)}, ..., f^{(K)}]^T$ and takes as input the matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]^T$ which contains the vector parameters $\mathbf{w}_k$ used in modelling the density ratios. We also modify the sigma function to $\sigma(\mathbf{f}, \mathbf{t}) = \frac{\exp\langle \mathbf{f}, \mathbf{t} \rangle}{\sum_k \exp f^{(k)}}$.

C Support Vector Machines

Thickness of Error Margin:

In this problem we have a tube defined by two boundaries: $f(\mathbf{x}; \mathbf{w}) = 1$ and $f(\mathbf{x}; \mathbf{w}) = -1$. The center line for this tube is given by $f(\mathbf{x}; \mathbf{w}) = 0$ suppose a point $\mathbf{x}_0$ lies on this center line, then the distance between this point and $f(\mathbf{x}; \mathbf{w}) = 1$ is

$$\frac{|\mathbf{w}'^T \mathbf{x}_0 - 1|}{||\mathbf{w}'||} = \frac{1}{||\mathbf{w}'||}$$

C.1 Soft Margin Classifier

KKT Conditions:

For a constrained problem of the form

$$\begin{aligned} &\text{Minimise } f(\mathbf{x}) \\ &\text{Subject to } g_i(\mathbf{x}) \geq 0 \text{ for } i = 1, ..., n \end{aligned}$$

The Karush-Kuhn-Tucker conditions are

(a) $\nabla_{\mathbf{x}} f(\hat{\mathbf{x}}) + \hat{\boldsymbol{\lambda}} \nabla_{\mathbf{x}} g(\hat{\mathbf{x}}) = 0$

(b) $g(\hat{\mathbf{x}}) \leq 0$

(c) $\hat{\boldsymbol{\lambda}} \geq 0$

(d) $\hat{\boldsymbol{\lambda}} g(\hat{\mathbf{x}}) = 0$

Using Lagrangian Dual to Rewrite Optimisation Problem

We can then get the Lagrangian dual as

$$\ell(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = \min_{\mathbf{w}, \boldsymbol{\epsilon}} ||\mathbf{w}'||^2 + \sum_i \epsilon_i - \lambda_i \left[y_i \left(\langle \mathbf{x}_i, \mathbf{w}' \rangle + w_0 \right) + \epsilon_i - 1 \right] - \lambda_i' \epsilon_i$$

Since the original problem is convex, the KKT conditions (provided in Appendix C.1) are sufficient to obtain an optimal solution. Under condition (a), we obtain the optimality conditions

$$\begin{aligned}\nabla_{\mathbf{w}'}\ell(\boldsymbol{\lambda}) = 2\mathbf{w}' - \sum_i \lambda_i y_i \mathbf{x}_i = 0 \implies \mathbf{w}' = \frac{1}{2}\sum_i \lambda_i y_i \mathbf{x}_i \\ \nabla_{\epsilon_i}\ell(\boldsymbol{\lambda}) = 1 - \lambda_i - \lambda_i' = 0 \implies \lambda_i + \lambda_i' = 1 \\ \nabla_{f_{\mathbf{w}}(\mathbf{x}_i)}\ell(\boldsymbol{\lambda}) = \sum_i \lambda_i y_i = 0\end{aligned}$$

We can also rewrite

$$\begin{aligned}||\mathbf{w}'|| &= \left\langle \frac{1}{2}\sum_i \lambda_i y_i \mathbf{x}_i, \frac{1}{2}\sum_j \lambda_j y_j \mathbf{x}_j \right\rangle \\ &= \frac{\sum_i \sum_j \lambda_i y_i \lambda_j y_j \mathbf{x}_i^T \mathbf{x}_j}{4} = \frac{\tilde{\boldsymbol{\lambda}}^T \mathbf{X}^T \mathbf{X} \tilde{\boldsymbol{\lambda}}}{4}\end{aligned}$$

as well as

$$\begin{aligned}\sum_i \epsilon_i - \sum_i \lambda_i \epsilon_i - \sum_i \lambda_i' \epsilon_i = \sum_i \epsilon(1 - \lambda_i - \lambda_i') = 0 \\ -\sum_i \lambda_i y_i f_{\mathbf{w}}(\mathbf{x}_i) = -\sum_i \lambda_i y_i \mathbf{w}'^T \mathbf{x}_i - \sum_i \lambda_i y_i w_0 \\ = -\sum_i \lambda_i y_i \left(\frac{1}{2}\sum_j \lambda_j y_j \mathbf{x}_j\right) = -\frac{\tilde{\boldsymbol{\lambda}}^T \mathbf{X}^T \mathbf{X} \tilde{\boldsymbol{\lambda}}}{2}\end{aligned}$$

where $\tilde{\boldsymbol{\lambda}} = [\lambda_1 y_1, \ldots, \lambda_n y_n]$.

So for $\boldsymbol{\lambda}, \mathbf{w}, \boldsymbol{\epsilon}$ satisfying the KKT optimatility conditions, the Lagrangian becomes

$$\ell(\boldsymbol{\lambda}) = -\frac{\tilde{\boldsymbol{\lambda}}^T \mathbf{X}^T \mathbf{X} \tilde{\boldsymbol{\lambda}}}{4} + \langle \boldsymbol{\lambda}, \mathbf{1} \rangle$$

equivalently

$$\begin{aligned}\hat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} -\frac{\tilde{\boldsymbol{\lambda}}^T \mathbf{X}^T \mathbf{X} \tilde{\boldsymbol{\lambda}}}{4} + \langle \boldsymbol{\lambda}, \mathbf{1} \rangle \\ \text{Subject to } \lambda_i \in [0, 1] \\ \sum_i \lambda_i y_i = 0\end{aligned}$$

and so we take $\hat{\mathbf{w}}' = \frac{\sum_i \hat{\lambda}_i y_i \mathbf{x}_i}{2}$ and all that's left is to find $\hat{w}_0$. We consider three possible cases, using the fact $\lambda_i(y_i f_{\mathbf{w}}(\mathbf{x}_i) + \epsilon_i - 1) = 0$:

Case 1: $(\mathbf{x}_i, y_i)$ is on the correct side of the tube:

$\implies y_i f_{\mathbf{w}}(\mathbf{x}_i) > 1 \implies \epsilon_i = 0$, so we must have $\lambda_i = 0$

Case 2: $(\mathbf{x}_i, y_i)$ is on the wrong side of the tube:

$\implies y_i f_{\mathbf{w}}(\mathbf{x}_i) < 1 \implies \epsilon_i > 0$, so $\lambda_i' = 0$ and $\lambda_i = 1$

Case 3: $(\mathbf{x}_i, y_i)$ is on the boundary of the tube:

$\implies y_i f_{\mathbf{w}}(\mathbf{x}_i) = 1 \implies \epsilon_i = 0$ and so $\lambda_i \in [0, 1]$.
Then we also see $y_i f_{\mathbf{w}}(\mathbf{x}_i) - 1 = 0 \implies w_0 = \frac{1}{y_i} - \langle \mathbf{x}_i, \mathbf{w}' \rangle$

Further we can write $\mathbf{K} = \mathbf{X}^T\mathbf{X}$ so that we can use kernel functions $k(\mathbf{x}_i, \mathbf{x}_j)$. Hence our decision function becomes:

$$f(\mathbf{x}; \mathbf{w}) = \mathbf{w}'^T\mathbf{x} + w_0 = \frac{1}{2}\sum_i \hat{\lambda}_i y_i \mathbf{x}_i^T\mathbf{x} + w_0 = \frac{1}{2}\sum_i \hat{\lambda}_i y_i k(\mathbf{x}_i, \mathbf{x}) + w_0$$

Thus our objective function is

$$\min_{\mathbf{w}} ||\mathbf{w}'||^2 + \begin{cases} 0 & \text{if } y_i f_{\mathbf{w}}(\mathbf{x}_i) \geq 0 \\ 1 - y_i f_{\mathbf{w}}(\mathbf{x}_i) & \text{if } y_i f_{\mathbf{w}}(\mathbf{x}_i) < 0 \end{cases}$$