1 Linear Methods for Classification

In binary classification we saw that decisions are made by checking the sign of a linear decision function $f(\mathbf{x})$. In multiclass classification this no longer suffices. Naïve methods involving the use of multiple binary classifiers combined using some rule (e.g. majority voting) fail to reliably cover the whole space and provide unambiguous predictions due to the more complicated geometry (intersection of hyperplanes). Instead, we choose to fit K decision functions $\{f_k(\mathbf{x}; \mathbf{w}_k)\}_{k=1}^K$, and predict labels by $\hat{k} = \max_k f_k(\mathbf{x}; \mathbf{w}_k)$. This no longer has a simple geometric interpretation, but can be interpreted probabilistically.

1.0.1 Least Squares (LS) Classifier

To predict the decision function f we can use the same least squares approach as we have in the regression problem, only now $y = \{1, -1\}$ rather than a continuous value and our predictions take the form $\hat{y} = \text{sign}(f(\mathbf{x}; \mathbf{w}_{LS}))$. We can also use feature transforms $\boldsymbol{\phi}(\mathbf{x})$ as we did for the regression problem which can enable us to classify data that was unseparable in the original space, but is separable in the feature space.

LS classification can be extended to the multiclass case using *one-hot encoding* where we replace the set of labels $y = \{1, \dots, K\}$ with a vector $\boldsymbol{t}_i^T \in R^K$ populated with 0's and a single 1 in the k^{th} slot corresponding to $y_i = k$. We solve the LS problem by

$$\underset{\boldsymbol{W}}{\text{argmin}} \sum_{i \in D} \left\| \boldsymbol{t}_i^T - \boldsymbol{W}^T \tilde{\mathbf{x}} \right\|^2$$

with $\boldsymbol{W}_{LS} \in R^{(d+1)\times K}$ and $\tilde{\mathbf{x}}_i = [\mathbf{x}_i, 1]^T \in R^{d+1}$.

The LS classifier has some downsides however, in that it is highly sensitive to data far from the bulk which will skew the decision boundary and the model lacks any probabilistic interpretation (unlike the regression formulation which can be derived as the MLE of a probabilistic model).

1.0.2 Fisher Discriminant Analysis

An alternative method is to project the input space onto a 1-dimensional space by taking the inner product with the weight vector, i.e. $\langle \mathbf{w}, \mathbf{x} \rangle$. We then want to try to maximize the class separation and minimize the within class variance in the projected space. Note that in this formulation, the inner product does not define a decision function and we classify points by thresholding the projected space (e.g $y(\mathbf{x}) \geq y_0 \rightarrow C_1$, else C_2). This also means that the algorithm makes no attempt to minimize false positives or negatives. For the 2-class problem, we find the solution for the projection as $\mathbf{w} = \boldsymbol{S}_{\mathbf{w}}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)$ with $\boldsymbol{S}_{\mathbf{w}}$ as the within class variance and $\boldsymbol{\mu}_\pm$ the class means. A.1.

1.1 Probabilistic Generative Classifiers

To build a probabilistic classifier, we minimize the expected loss: $\text{argmin}_{y_0} \text{E}_{p(y \mid \mathbf{x})}[L(y, y_0)|\mathbf{x}]$. To build a generative model, we estimate $p(\mathbf{x} \mid y)$

1.1.1 Continuous Input Variable

If $\mathbf{x}$ is continous we might model $p(\mathbf{x} \mid \mathbf{y})$ with a MVN.

Linear Decision Boundary If we assume that our $\mathbf{x}$ are modelled by a MVN with shared covariance, we find that the decision boundary we get is piecewise-linear (see A.3). To actually make predictions we need to estimate the MVN parameters, which we can do with MLE:

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{i \in D, y_i = k} \mathbf{x}_i$$

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

If we relax the restriction that the variables share the same covariance, then we no longer have a linear decision boundary.

1.1.2 Discrete Input Variable

If $\mathbf{x}$ is discrete, we will want to model it in a different way. In the case where $\mathbf{x}$ is a "bag of words" representation, each $\mathbf{x}_i \in N^d$ is a vector representing the frequency that words in some overall dictionary (of size d) appear in the i^{th} document. e.g. if we have the document "to be or not to be" with dictionary $\{to, be, or, not, question\}$ then we would have $\mathbf{x} = [2, 2, 1, 1, 0]^T$.

Naïve Bayes One approach to this task is to use a naïve Bayes model which assumes that each element of $\mathbf{x}$ is conditionally independent given y. Hence we can use a multinomial distribution

$$p(\mathbf{x} = \mathbf{x}_0 \mid y) \propto \prod_{i=1}^{d} p(x^{(i)} \mid y)^{x_0^{(i)}}$$

We can estimate the conditional and prior probabilities using Maximum Likelihood Estimation (A.2) and compute predictions as

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(\mathbf{x} = \mathbf{x}_0 \mid y) p(y)$$

1.2 Discriminative Classifiers

Taking a discriminative approach (i.e. estimating $p(y \mid \mathbf{x})$ directly) we want to make a model assumption $p(y \mid \mathbf{x}; \mathbf{w})$, construct a likelihood function $p(D \mid \mathbf{w})$ and estimate the parameters $\mathbf{w}$.

If we take a binary classification model and use Bayes Rule and rearrange, we find that we can represent the conditional probability of y using a density ratio:

$$p(y = \pm 1 \mid \mathbf{x}) = \left(1 + \frac{p(\mathbf{x} \mid y = \mp) p(y = \mp)}{p(\mathbf{x} \mid y = \pm) p(y = \pm)}\right)^{-1}$$

in contrast to the generative approach which requires estimating the class density $p(\mathbf{x} \mid y)$ itself. Thus discriminative models need only make assumptions about this ratio.

Hence we can now define a model

$$\pm f(\mathbf{x};\mathbf{w}) = \log \frac{p(\mathbf{x} \mid y = \mp)p(y = \mp)}{p(\mathbf{x} \mid y = \pm)p(y = \pm)}$$

and a corresponding activation function $\sigma(t) = (1 + \exp(t))^{-1}$ s.t. $p(y \mid \mathbf{x};\mathbf{w}) = \sigma(y \cdot f(\mathbf{x};\mathbf{w}))$.

If we take $f(\mathbf{x};\mathbf{w}) = \langle \mathbf{w}, \mathbf{x} \rangle$ (i.e. to be a linear model), then we have a *logistic regression*[1] model. This model formulation is more robust to outliers than the least squares model due to the "squashing" effect of the *sigmoidal* activation function shape. In this framework we can also introduce feature transforms as we did in the least squares model.

In the binary logistic regression problem, we find the decision function assigns predicted classes at the decision boundary where $p(y \mid \mathbf{x};\mathbf{w}) = \frac{1}{2}$. A.4.1.

Now, as before, we need to estimate the parameters $\mathbf{w}$ which we can do by

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

MAP $\mathbf{w}_{MAP} = \text{argmax}_{\mathbf{w}} \sum_{i \in D} \log \sigma(f(\mathbf{x}_i;\mathbf{w}) \cdot y_i) + \log p(\mathbf{w})$

Full probabilistic

$$\begin{aligned} p(y \mid \mathbf{x}) &= \int p(y \mid \mathbf{x};\mathbf{w}) p(\mathbf{w} \mid D) d\mathbf{w} \\ &\propto \int \sigma(f(\mathbf{x};\mathbf{w}) \cdot y) \prod_{i \in D} \sigma(f(\mathbf{x}_i;\mathbf{w}) \cdot y_i) p(\mathbf{w}) d\mathbf{w} \end{aligned}$$

Note that the full probabilistic calculation cannot be done in closed form and the maximization problems also need to be solved numerically.

1.2.1 Multiclass Logistic Regression

We can write down our conditional probability in a similar way to the binary classification case as

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

and hence model the log ratio

$$\log \frac{p(\mathbf{x} \mid y = k')p(y = k')}{p(\mathbf{x} \mid y = k)p(y = k)} = f(\mathbf{x};\mathbf{w}_{k'},\mathbf{w}_k) = g(\mathbf{x};\mathbf{w}_{k'}) - g(\mathbf{x};\mathbf{w}_k)$$

Using one-hot encoding again (as in 1.0.1) we define $\boldsymbol{f}(\mathbf{x};\mathbf{w}) = \boldsymbol{W}^T \tilde{\mathbf{x}}$ and calculate MLE as

$$\underset{\mathbf{w}}{\text{argmax}} \sum_{i \in D} \log \sigma(\boldsymbol{f}(\mathbf{x}_i;\mathbf{w}), \boldsymbol{t}_i)$$

with $\sigma(\boldsymbol{f}, \boldsymbol{t}) = \frac{\exp\langle \boldsymbol{f}, \boldsymbol{t} \rangle}{\sum_k \exp f^{(k)}}$ (A.4).

1.3 Support Vector Machines (SVM)

A "good" classifier should minimize the error on unseen data (not just the training set), i.e. it should be able to generalize. Based on this, if we take a class boundary

[1] logistic regression is really a classification algorithm

that misclassifies when a small perturbation is applied to the data, we say that it has a small *margin of error*. In a linear model, the width of the margin is given by $\frac{1}{\|\mathbf{w}'\|}$ (A.5.1). To be able to generalize succesfully, we want to maximize this margin when we build our classifier, whilst maintaining the correct labelling of our training points. This means our problem is one of constrained minimization (note that we can easily transform a maximization problem into a minimization):

$$\begin{aligned} \min_{\mathbf{w}'} \quad & \left\|\mathbf{w}'\right\|^2 \\ \text{subject to} \quad & y_i f(\mathbf{x}_i; \mathbf{w}) \geq 1 \quad \forall i \end{aligned}$$

Effectively we increase the width of our margin until we hit the edge of the class sets.

1.3.1 Soft Margin Classifier

In the case of overlapping classes we are guaranteed to have misclassifications within any margin. In this case we need to replace our *hard* constraints with *soft* ones which we do by introducing a small positive compensation that can move the points on the wrong side of our boundary to the correct side. We want to keep the compensation as small as possible, to make as few misclassifications as possible (which means that the solution for ϵ should be sparse).

$$\begin{aligned} \min_{\mathbf{w}', \boldsymbol{\epsilon}} \quad & \left\|\mathbf{w}'\right\|^2 + \sum_i \epsilon_i \\ \text{subject to} \quad & y_i f(\mathbf{x}_i; \mathbf{w}) + \epsilon_i \geq 1, \quad \epsilon_i \geq 0, \quad \forall i \end{aligned}$$

Note that in the case of a linear model, this is a *convex* optimization problem, since the objective function is the sum of a norm and linear function, which is convex, and the inequalities are a set of halfspaces (and linear functions), which are convex.

1.3.2 Lagrangian Dual

To solve our constrained minimization problem, we use the method of *Lagrange Multipliers* and calculate the *dual* of our original (*primal*) problem. Note that for this problem, being both convex and satisfying *Slater's condtion*, *strong duality* holds, so the solutions to the dual and primal problems are equal. We end up with (A.5.3)

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & -\frac{1}{4}\tilde{\boldsymbol{\lambda}}^T \mathbf{X}\mathbf{X}^T \tilde{\boldsymbol{\lambda}} + \langle \mathbf{1}, \boldsymbol{\lambda} \rangle \\ \text{subject to} \quad & 0 \leq \lambda_i \leq 1, \quad \sum_i \lambda_i y_i = 0 \end{aligned}$$

which we note differs to the original problem in a few ways: the optimization variable; it has simpler constraints; the dimensionality now scales with n (vs d); and that we can make use of the kernel trick ($\mathbf{K} = \mathbf{X}\mathbf{X}^T$). The margin boundaries always pass through some data points, which defines the extent of the margin and these points are called *support vectors*. (The margins now form *supporting hyperplanes* of the class sets). Downsides to SVMs are the high computational cost associated with the constrained optimization, the lack of a probabilistic interpretation and difficulties in implementing a multi-class version.

Appendices

A Appendix

A.1 Fisher Discriminant Analysis

Projected centre for class k and overall dataset:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i,y_i=k} \mathbf{w}^T \mathbf{x}_i$$

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{w}^T \mathbf{x}_i$$

and within class scatterness/variance and between class scatterness/variance:

$$s_{w,k} = \sum_{i,y_i=k} (\mathbf{w}^T \mathbf{x}_i - \hat{\mu}_k)^2$$

$$s_{b,k} = n_k (\hat{\mu}_k - \hat{\mu})^2$$

n_k is needed to make $s_{b,k}$ the same scale as $s_{w,k}$.

So we want to maximize between class scatterness and minimize within class scatterness $\forall k$, i.e.

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

whose solution for $K = 2$ is given by

$$\mathbf{w} = \boldsymbol{S}_{\mathbf{w}}^{-1} (\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)$$

$$\boldsymbol{S}_{\mathbf{w}} = \sum_{k=1}^{K} \boldsymbol{S}_k$$

where $\boldsymbol{S}_k$ is the sample covariance for class k times n_k. See [1] 4.14 for the full derivation.

A.2 Multinomial

Deriving the MLEs of multinomial distribution parameters. Denote the PMF of the Multinomial distribution as $\pi(x_1, \ldots, x_K; n, p_1, \ldots, p_K)$ with frequencies of event k given by x_k, total number of trials given by n and probability of each even event given by p_k. So we start by writing out the PMF, then we take logs and maximize w.r.t the parameters, adding a constraint that the parameters sum to 1 (since they

are probabilities), using Lagrange multipliers, λ:

$$\pi(x_1, \dots, x_K; n, p_1, \dots, p_K) = \frac{n!}{x_1! \dots x_K!} p_1^{x_1} \dots p_K^{x_K}$$

$$L = \log n! - \sum_{k=1}^{K} \log x_k! + \sum_{k=1}^{K} x_k \log p_k + \lambda(1 - \sum_{k=1}^{K} p_k)$$

$$\partial_{p_i} \log \pi = 0 = \frac{x_i}{p_i} - \lambda$$

$$\sum_{i=1}^{K} x_i = \sum_{i=1}^{K} p_i \lambda$$

$$\implies \lambda = n$$

$$\implies p_i = \frac{x_i}{n}$$

So the MLE of the probability of event k is the ratio of the number of occurrences of k to the total number of events n.

From this we can see that if we assume a Multinomial prior over y we have

$$p(y) \propto \prod_{k=1}^{K} p(y = k)^{n_k}$$

$$\implies p(y = k) = \frac{n_k}{n}$$

Similarly, if we take a Multinomial Likelihood for the conditional distribution we have, for a particular point $\mathbf{x}_0$

$$p(\mathbf{x} = \mathbf{x}_0 \mid y) \propto \prod_{i=1}^{d} p(x^{(i)} \mid y)^{x_0^{(i)}}$$

$$\implies p(x^{(i)} \mid y = k) = \frac{\sum_{j \in D, y_j = k} x_j^{(i)}}{\sum_{j \in D, y_j = k} \sum_{k=1}^{d} x_j^{(i)}}$$

$$= \frac{n_{ik}}{n_k}$$

which is now the frequency at which the word $x^{(i)}$ occurs in class k.

A.3 Linear Decision Boundary

With a shared covariance matrix MVN model, prove that the decision boundary is piecewise-linear. The decision boundary between two classes $B_{kk'}$ is given by the set of $\mathbf{x}$ where the probability of belonging to two different classes is equal;

$$B_{kk'} = \{\mathbf{x} \mid p(y = k \mid \mathbf{x}; \hat{\mathbf{w}}) = p(y = k' \mid \mathbf{x}; \hat{\mathbf{w}}), k \neq k'\} \tag{1}$$

$$= \left\{\mathbf{x} \,\middle|\, \frac{p(\mathbf{x} \mid y = k; \hat{\mathbf{w}}) p(y = k)}{p(\mathbf{x} \mid y = k'; \hat{\mathbf{w}}) p(y = k')} = 1, k \neq k'\right\} \tag{2}$$

$$= \{\mathbf{x} \mid \log p(\mathbf{x} \mid y = k; \hat{\mathbf{w}}) + \log p(y = k) - \log p(\mathbf{x} \mid y = k'; \hat{\mathbf{w}}) + \log p(y = k') = 0\} \tag{3}$$

$$= \{\mathbf{x} \mid -(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) + (\mathbf{x} - \boldsymbol{\mu}_{k'})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{k'}) = C\} \tag{4}$$

$$= \{\mathbf{x} \mid 2(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'})^T \boldsymbol{\Sigma}^{-1} \mathbf{x} = C\} \tag{5}$$

where we have absorbed all of the constant terms into C on the right handside (which we have redefined to include the term quadratic in $\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'}$ in the last line). We can see that for each pair of classes, k, k' we define an affine set as a boundary.

A.3.1 MVN

Prove that if $p(\mathbf{x}\,|\,y = \pm 1) \sim N_{\mathbf{x}}(\boldsymbol{\mu}_\pm, \boldsymbol{\Sigma})$ then $\exists \mathbf{w}^*$ s.t. $p(y|\mathbf{x}) = \sigma((\langle \mathbf{x}, \mathbf{w}^* \rangle + w_0^*)y)$. First, write out the conditional of y on $\mathbf{x}$ using Bayes Rule and then rearrange the terms.

$$
\begin{aligned}
p(y \mid \mathbf{x}) &= \frac{p(\mathbf{x} \mid y)p(y)}{p(\mathbf{x})} \\
p(y = \pm 1 \mid \mathbf{x}) &= \frac{N_{\mathbf{x}}(\boldsymbol{\mu}_\pm, \boldsymbol{\Sigma})p(y = \pm 1)}{N_{\mathbf{x}}(\boldsymbol{\mu}_+, \boldsymbol{\Sigma})p(y = 1) + N_{\mathbf{x}}(\boldsymbol{\mu}_-, \boldsymbol{\Sigma})p(y = -1)} \\
&= \frac{\mathrm{e}^{-t_\pm} p(y = \pm 1)}{\mathrm{e}^{-t_+} p(y = 1) + \mathrm{e}^{-t_-} p(y = -1)} \\
&= \frac{p(y = \pm 1)}{\mathrm{e}^{-t_+ + t_\pm} p(y = 1) + \mathrm{e}^{-t_- + t_\pm} p(y = -1)} \qquad = \frac{1}{\mathrm{e}^{-t_+ + t_\pm} \frac{n_+}{n_\pm} + \mathrm{e}^{-t_- + t_\pm} \frac{n_-}{n_\pm}}
\end{aligned}
$$

where we have assumed that we can approximate $p(y = \pm 1) = \frac{n_\pm}{n}$ and rearrange in the final line. Now looking at only the exponential terms and simplifying:

$$
-t_- + t_\pm = \underbrace{\mp(\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_- - \boldsymbol{\mu}_+) \pm \log \frac{n_-}{n_+}}_{w_0^*} \pm \underbrace{2(\boldsymbol{\mu}_- - \boldsymbol{\mu}_+)^T \boldsymbol{\Sigma}^{-1}}_{\mathbf{w}^{*T}} \mathbf{x}
$$

so we see that we produce the desired result.

A.4 Logistic Regression

From the multiclass conditional probability, we can define a log-ratio model and hence a multiclass activation (sigmoidal) function.

$$
\begin{aligned}
p(y = k \mid \mathbf{x}) &= \frac{p(\mathbf{x} \mid y = k)p(y = k)}{\sum_{k'} p(\mathbf{x} \mid y = k')p(y = k')} \\
&= \frac{p(\mathbf{x} \mid y = k)p(y = k)}{p(\mathbf{x} \mid y = k_0)p(y = k_0)} \left(\sum_{k'} \frac{p(\mathbf{x} \mid y = k')p(y = k')}{p(\mathbf{x} \mid y = k_0)p(y = k_0)} \right)^{-1} \\
&= \exp(f(\mathbf{x}; \mathbf{w}_k)) \left(\sum_{k'} \exp(f(\mathbf{x}; \mathbf{w}_{k'})) \right)^{-1} \\
&= \frac{\mathrm{e}^{\langle \boldsymbol{f}, \boldsymbol{t} \rangle}}{\sum_k \mathrm{e}^{f^{(k)}}}
\end{aligned}
$$

where we have defined some reference class k_0 s.t. we can subsequently define our log-ratio model as

$$
f^{(k)} = f(\mathbf{x}; \mathbf{w}_k) = \log \frac{p(\mathbf{x} \mid y = k)p(y = k)}{p(\mathbf{x} \mid y = k_0)p(y = k_0)}
$$

and using the one-hot encoded target vector $t_i = \begin{cases} 1 & i = k \\ 0 & \text{otherwise} \end{cases}$.

A.4.1 Binary Logistic Regression

Find the decision boundary (B) of a binary logistic regression problem:

$$\begin{aligned} B &= \{\mathbf{x} \mid p(y=+1 \mid \mathbf{x}; \mathbf{w}) = p(y=-1 \mid \mathbf{x}; \mathbf{w})\} \\ &= \{\mathbf{x} \mid (1+\mathrm{e}^{\mathbf{w}^T\mathbf{x}})^{-1} = (1+\mathrm{e}^{-\mathbf{w}^T\mathbf{x}})^{-1}\} \\ &= \{\mathbf{x} \mid \mathbf{w}^T\mathbf{x} = -\mathbf{w}^T\mathbf{x}\} \\ \implies & \mathbf{x} = 0 \\ \implies & p(y=+1 \mid 0; \mathbf{w}) = (1+\mathrm{e}^0)^{-1} = \frac{1}{2} \end{aligned}$$

A.5 SVM

A.5.1 Margin Width

Finding the width of the margin between the hyperplane at $y(\mathbf{x}) = 0$ and $y(\mathbf{x}) = 1$: The hyperplanes run parallel to one another and perpendicular to the vector $\mathbf{w}'$. So to find their distance, we can pick a pair of points in the two sets that is separated by a vector parallel to $\mathbf{w}'$. Taking $f(\mathbf{x}; \mathbf{w}) = \langle \mathbf{w}', \mathbf{x} \rangle + w_0$.

$$\begin{aligned} H_0 &= \{\mathbf{x} \mid \mathbf{w}'^T\mathbf{x} + w_0 = 0\} \\ H_1 &= \{\mathbf{x} \mid \mathbf{w}'^T\mathbf{x} + w_0 = 1\} \\ \mathbf{w}'^T\mathbf{x}_0 + w_0 &= 0 \\ \mathbf{w}'^T\mathbf{x}_1 + w_0 &= 1 \\ \mathbf{w}'^T(\mathbf{x}_1 - \mathbf{x}_0) &= 1 \\ \left\|\mathbf{w}'^T(\mathbf{x}_1 - \mathbf{x}_0)\right\| &= 1 \\ \implies \|\mathbf{x}_1 - \mathbf{x}_0\| &= \frac{1}{\|\mathbf{w}'\|} \end{aligned}$$

where we have used that the Cauchy-Schwartz inequality is an equality when the vectors in the inner product run parallel (as they do here, by construction).

A.5.2 Lagrangian Dual

Given a constrained problem

$$\begin{aligned} \min_{\theta} \quad & f_0(\theta) \\ \text{subject to} \quad & f_i(\theta) \leq 0 \end{aligned}$$

we can construct the *Lagrangian* function L and *dual* function g:

$$L(\theta, \boldsymbol{\lambda}) = f_0(\theta) + \sum_i \lambda_i f_i(\theta)$$

$$g(\boldsymbol{\lambda}) = \min_{\theta} L(\theta, \boldsymbol{\lambda})$$

where we call the $\lambda_i \geq 0$ the *Lagrange Multipliers*.

A.5.3 Soft-Margin Classifiers

Using the defintions in A.5.2 we write down the Lagrange dual function for the soft-margin classifier:

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

we calculate the optimality conditions by differentiating w.r.t the parameters of interest, as usual:

$$\begin{aligned}
\partial_{\mathbf{w}'} L(\mathbf{w}, \boldsymbol{\epsilon}, \boldsymbol{\lambda}, \boldsymbol{\lambda}') = 0 = 2\mathbf{w}' - \sum_i \lambda_i y_i \mathbf{x}_i \\
\implies \hat{\mathbf{w}}' = \frac{1}{2}\sum_i \lambda_i y_i \mathbf{x}_i = \frac{1}{2}\mathbf{X}^T\tilde{\boldsymbol{\lambda}} \\
\partial_{\boldsymbol{\epsilon}} L = 0 = \mathbf{1} - \boldsymbol{\lambda} - \boldsymbol{\lambda}' \\
\implies 1 = \lambda_i + \lambda_i' \\
\partial_{w_0} L = 0 = -\sum_i \lambda_i y_i
\end{aligned}$$

where we have defined $\tilde{\boldsymbol{\lambda}} := \mathbf{y} \odot \boldsymbol{\lambda}$. Now we can write down the dual function (by substituting the optimal conditions above into the Lagrangian function) as:

$$\begin{aligned}
g(\boldsymbol{\lambda}) &= \left\|\frac{1}{2}\mathbf{X}^T\tilde{\boldsymbol{\lambda}}\right\|^2 + \mathbf{1}^T\boldsymbol{\epsilon} - \frac{1}{2}\tilde{\boldsymbol{\lambda}}^T\mathbf{X}\mathbf{X}^T\tilde{\boldsymbol{\lambda}} - w_0\mathbf{1}^T\tilde{\boldsymbol{\lambda}} - \boldsymbol{\lambda}^T\boldsymbol{\epsilon} + \mathbf{1}^T\boldsymbol{\lambda} - \boldsymbol{\lambda}'^T\boldsymbol{\epsilon} \\
&= \frac{1}{4}\tilde{\boldsymbol{\lambda}}^T\mathbf{X}\mathbf{X}^T\tilde{\boldsymbol{\lambda}} - \frac{1}{2}\tilde{\boldsymbol{\lambda}}^T\mathbf{X}\mathbf{X}^T\tilde{\boldsymbol{\lambda}} + \mathbf{1}^T\boldsymbol{\epsilon} - (\boldsymbol{\lambda} + \boldsymbol{\lambda}')^T\boldsymbol{\epsilon} + \mathbf{1}^T\boldsymbol{\lambda} \\
&= -\frac{1}{4}\tilde{\boldsymbol{\lambda}}^T\mathbf{X}\mathbf{X}^T\tilde{\boldsymbol{\lambda}} + \langle \mathbf{1}, \boldsymbol{\lambda}\rangle
\end{aligned}$$

Finally, we are left with the dual problem

$$\begin{aligned}
\max_{\boldsymbol{\lambda}} \quad & -\frac{1}{4}\tilde{\boldsymbol{\lambda}}^T\mathbf{X}\mathbf{X}^T\tilde{\boldsymbol{\lambda}} + \langle \mathbf{1}, \boldsymbol{\lambda}\rangle \\
\text{subject to} \quad & 0 \leq \lambda_i \leq 1, \quad \sum_i \lambda_i y_i = 0
\end{aligned}$$

Also note that we can rewrite our decision function $f(\mathbf{x}; \mathbf{w})$ using our solution for $\mathbf{w}'$ as

$$\begin{aligned}
f(\mathbf{x}; \mathbf{w}) &= \langle \mathbf{w}', \mathbf{x}\rangle + w_0 \\
&= \frac{1}{2}\tilde{\boldsymbol{\lambda}}^T\mathbf{X}\mathbf{x} + w_0 \\
&= \frac{1}{2}\tilde{\boldsymbol{\lambda}}^T\mathbf{k} + w_0
\end{aligned}$$

with $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ and $\mathbf{k}(\mathbf{x}) = k(\mathbf{X}, \mathbf{x}) = \mathbf{X}\mathbf{x}$.

A.5.4 Connection to Logistic Regression

By further analysing the soft-margin SVM, we can find a connection to the Logistic Regression model we looked at. We can consider the minimum required compensation ϵ_i to fix a given misclassification as $\epsilon_i = 1 - y_i f(\mathbf{x}_i)$ (i.e. to set the inequality constraint to 0). With this definition for ϵ we can define a function to describe

this behaviour as the hinge function $h(t) = \begin{cases} 0 & t \geq 1 \\ 1-t & t < 1 \end{cases}$. This function is not differentiable though, so we can choose to approximate it with a smooth upper bound: $h_{smooth}(t) = -\log \frac{1}{1+e^{-t}} = -\log \sigma(-t)$. This allows us to reforumlate our optimization problem as the unconstrained problem:

$$\min_{\mathbf{w}} \quad \left\|\mathbf{w}'\right\|^2 - \sum_i \log \sigma(-y_i f(\mathbf{x}_i; \mathbf{w}))$$

which we can transform to the equivalent maximization problem

$$\max_{\mathbf{w}} \quad \sum_i \log \sigma(-y_i f(\mathbf{x}_i; \mathbf{w})) - \left\|\mathbf{w}'\right\|^2$$

which looks very similar to the regularized logistic regression problem (as seen in 1.2.1).

A.5.5 Least Squares Optimization

Given the optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \sum_i (y_i - f(\mathbf{x}_i; \mathbf{w}))^2 \\ \text{subject to} \quad & \left\|\mathbf{w}'\right\|^2 \leq c \end{aligned}$$

we can again use Lagrange multipliers to construct the dual problem. Beginning by writing down the Lagrangian, and then computing the derivatives w.r.t $\mathbf{w}$ and w_0:

$$\begin{aligned} L(\mathbf{x}, \mathbf{w}, \lambda) &= \sum_i (y_i - f(\mathbf{x}_i; \mathbf{w}))^2 + \lambda(\left\|\mathbf{w}'\right\|^2 - c) \\ \partial_{\mathbf{w}'} L = 0 &= \sum_i (\partial_{\mathbf{w}'} (\mathbf{w}'^T \mathbf{x}_i)^2 - 2 y_i \mathbf{x}_i) + 2\lambda \mathbf{w}' \\ \partial_{w'_a} w'_j x_{ji} w'_k x_{ki} &= 2 x_{ai} w'_k x_{ki} = 2(\mathbf{w}'^T \mathbf{x}_i)\mathbf{x}_i \\ 0 &= \sum_i ((\mathbf{w}'^T \mathbf{x}_i)\mathbf{x}_i - y_i \mathbf{x}_i) + \lambda \mathbf{w}' \\ (\lambda \mathbf{I} + \sum_i \mathbf{x}_i \mathbf{x}_i^T)\mathbf{w}' &= \mathbf{X}^T \mathbf{y} \\ \implies \mathbf{w}'_* &= (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y} \end{aligned}$$

where we have assumed that $f(\mathbf{x}; \mathbf{w}) = \langle \mathbf{w}', \mathbf{x} \rangle + w_0$ and we have used Einstein notation convention in line 3 calculating the derivative of the squared inner product. Thus we see that we recover the regularized least squares solution.

For the computation of w_0 we have:

$$\begin{aligned} \partial_{w_0} L = 0 &= \sum_i (w_0 - y_i) \\ \implies \hat{w_0} &= \frac{1}{n} \sum_i y_i \end{aligned}$$

Now, with these optimality criteria we can write down the dual function as

$$g(\lambda) = \sum_i (\langle \mathbf{w}'_*, \mathbf{x}_i \rangle)^2 + \lambda(\left\|\mathbf{w}'_*\right\|^2 - c)$$

References

[1] Christopher M Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, NY, 2006. Softcover published in 2016.