Machine Learning 1

Lecture 06 - Linear Classification

Patrick Forré

Linear and Quadratic Discriminant Analysis (LDA), (QDA)

- Let K classes $\{c_1, \ldots, c_K\}$ be given. Classify $x \in \mathbb{R}^D$.
- We will model the joint distribution $p(x, t) = p(x|t)p(t)$ of the data points x with class t.
- Since the prior $p(t)$ is given by just K values $p(c_1), \ldots, p(c_K)$ we are left to model $p(x|c_k)$ for $k = 1, \ldots, K$.
- Model assumption: all conditional distributions $p(x|c_k)$ are D-dimensional Gaussian:

$$
\begin{aligned}
p(x|c_k) &= \mathcal{N}(x|\mu_k, \Sigma_k) \\
&= \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)\right),
\end{aligned}
$$

- If the covariance matrices Σ_k are all equal, then this is called Linear Discriminant Analysis (LDA),
- otherwise Quadratic Discriminant Analysis (QDA).
- For minimizing e.g. misclassification or expected loss we need to estimate the posterior $p(c_k|x) = \frac{p(x|c_k)p(c_k)}{p(x)}$, or just the quotients $\frac{p(c_k|x)}{p(c_K|x)} = \frac{p(x|c_k)}{p(x|c_K)}\frac{p(c_k)}{p(c_K)}$ for $k = 1, \ldots, K - 1$.

Preliminary: Sigmoid and Softmax function

- For K classes $\{c_1, \dots, c_K\}$ we can write the posterior as:

$$
\begin{aligned}
p(c_k|x) &= \frac{p(x|c_k)p(c_k)}{\sum_{j=1}^{K} p(x|c_j)p(c_j)} = \frac{\exp[\ln(p(x|c_k)p(c_k))]}{\sum_{j=1}^{K} \exp\left[\ln\left(p(x|c_j)p(c_j)\right)\right]} \\
&= \frac{\exp(a_k)}{\sum_{j=1}^{K} \exp(a_j)} =: \sigma_k(a_1, \dots, a_K), \\
a_j &= \ln\left(p(x|c_j)p(c_j)\right)
\end{aligned}
$$

- $\sigma_k(a_1, \dots, a_K)$ is called softmax function. This comes from:
- If $a_k >> a_j$ then $\sigma_k(a_1, \dots, a_K) \approx 1$ and $\sigma_j(a_1, \dots, a_K) \approx 0$.
- For $K = 2$ and classes $\{c_1, c_0\}$ we can write:

$$
\begin{aligned}
p(c_1|x) &= \frac{p(x|c_1)p(c_1)}{p(x|c_1)p(c_1)+p(x|c_0)p(c_0)} = \frac{1}{1+\frac{p(x|c_0)p(c_0)}{p(x|c_1)p(c_1)}} \\
&= \frac{1}{1+\exp(-a)} =: \sigma(a)
\end{aligned}
$$

with $a = \ln\left(\frac{p(x|c_1)p(c_1)}{p(x|c_0)p(c_0)}\right)$

- $\sigma(a)$ is called the (logistic) sigmoid function.
- Its inverse is the logit function: $\mathrm{logit}(b) = \ln\left(\frac{b}{1-b}\right)$.

Sigmoid function

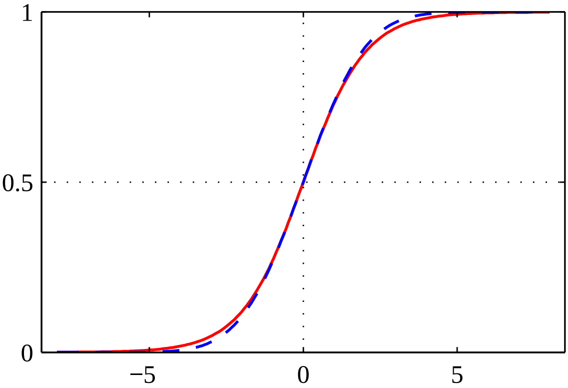

Figure: Sigmoid function $\sigma(a) = \frac{1}{1+\exp(-a)}$ (in red) and scaled cumulative normal distribution $\Phi(a) = \int_{-\infty}^{a} \mathcal{N}(x|0,1)dx$ (in blue). We have the symmetry property $\sigma(-a) = 1 - \sigma(a)$ and derivative $\sigma'(a) = \sigma(a)(1 - \sigma(a))$. (Bishop 4.9)

Linear Discriminant Analysis (LDA) for two classes

- We consider two classes $\{c_0, c_1\}$ with conditional distributions $p(x|c_k) = \mathcal{N}(x|\mu_k, \Sigma)$ with mean μ_k and fixed common covariance matrix Σ (LDA-model-assumption).
- Then the we get the log-ratios:

$$
\begin{aligned}
a &= \ln\left(\frac{p(x|c_1)}{p(x|c_0)}\frac{p(c_1)}{p(c_0)}\right) \\
&= \ln\mathcal{N}(x|\mu_1,\Sigma) - \ln\mathcal{N}(x|\mu_0,\Sigma) + \ln\left(\frac{p(c_1)}{p(c_0)}\right) \\
&= -\tfrac{1}{2}|\Sigma| - \tfrac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) \\
&\quad +\tfrac{1}{2}|\Sigma| + \tfrac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0) + \ln\left(\frac{p(c_1)}{p(c_0)}\right) \\
&= (\mu_1-\mu_0)^T\Sigma^{-1}x - \tfrac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \tfrac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 + \ln\left(\frac{p(c_1)}{p(c_0)}\right).
\end{aligned}
$$

- So we get the generalized linear form $p(c_1|x) = \sigma(w^Tx + w_0)$:

$$
\begin{aligned}
w &= \Sigma^{-1}(\mu_1-\mu_0), \\
w_0 &= -\tfrac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + \tfrac{1}{2}\mu_0^T\Sigma^{-1}\mu_0 + \ln\left(\frac{p(c_1)}{p(c_0)}\right),
\end{aligned}
$$

  where σ is the logistic sigmoid function.
- For prediction tasks we are left to fit the parameters on data.

Example: Linear Discriminant Analysis for two classes

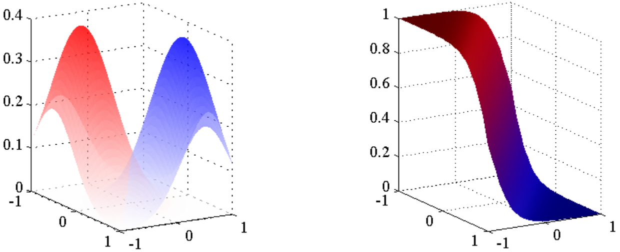

Figure: Left: Class conditional densities $p(x|c_i)$ for $x \in \mathbb{R}^2$. Right: Posterior $p(c_1|x)$ as sigmoid of linear function of x (Bishop 4.10)

Linear Discriminant Analysis (LDA) for multiple classes

- We consider two classes $\{c_1, \ldots, c_K\}$ with conditional distributions $p(x|c_k) = \mathcal{N}(x|\mu_k, \Sigma)$ with mean μ_k and fixed common covariance matrix Σ (LDA-assumption).
- With the use of the softmax function σ we get the form:

$$p(c_k|x) = \sigma_k(w_1^T x + w_{10}, \ldots, w_K^T x + w_{K0})$$

with the following weights:

$$\begin{aligned} w_j &= \Sigma^{-1}\mu_j, \\ w_{j0} &= -\tfrac{1}{2}\mu_j^T \Sigma^{-1}\mu_j + \ln p(c_j). \end{aligned}$$

- And we are left to fit the parameters.
- If every class c_k had its own covariance Σ_k we had to deal with a further quadratic terms in x (leading to Quadratic Discriminant Analysis QDA).

Linear Discriminant Analysis: Maximum Likelihood (I)

- Given: Data set $D = (x_1, \ldots, x_N)^T$ with binary classes $T = (t_1, \ldots, t_N)^T$ with $t_i \in \{c_0, c_1\} = \{0, 1\}$.
- Prior: $p(c_1) =: q$ and $p(c_0) = 1 - q$.
- If $t = 1$ we have $p(x, t) = p(x|t)p(t) = q \cdot \mathcal{N}(x|\mu_1, \Sigma)$.
- If $t = 0$ we have $p(x, t) = (1 - q) \cdot \mathcal{N}(x|\mu_0, \Sigma)$.
- This is summarized in one equation for the joint distribution:

$$p(x, t|q, \mu_0, \mu_1, \Sigma) = [q \cdot \mathcal{N}(x|\mu_1, \Sigma)]^t \cdot [(1 - q) \cdot \mathcal{N}(x|\mu_0, \Sigma)]^{1-t}$$

- Likelihood (for the training data under i.i.d. assumption):
  $p(D, T|q, \mu_0, \mu_1, \Sigma) =$

$$\prod_{n=1}^{N} [q \cdot \mathcal{N}(x_n|\mu_1, \Sigma)]^{t_n} \cdot [(1 - q) \cdot \mathcal{N}(x_n|\mu_0, \Sigma)]^{1-t_n} .$$

- Maximum Likelihood Estimator: Maximize the likelihood by taking the logarithm, then derivatives w.r.t. the parameters and putting the expression to zero. Solving for the $\frac{(D+1)(D+4)}{2}$ number of parameters then gives:

Linear Discriminant Analysis: Maximum Likelihood (II)

- For q we get:

$$q_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N} t_n = \frac{N_1}{N}, \qquad N_k := \#\{n | t_n = k\}$$

- For μ_0, μ_1 we get:

$$\mu_{1,\mathrm{ML}} = \frac{1}{N_1}\sum_{t_n=1} x_n, \qquad \mu_{0,\mathrm{ML}} = \frac{1}{N_0}\sum_{t_n=0} x_n,$$

- For Σ we get:

$$\begin{aligned}\Sigma_{\mathrm{ML}} &= \frac{N_1}{N}\left[\frac{1}{N_1}\sum_{t_n=1}^{N}(x_n - \mu_{1,\mathrm{ML}})(x_n - \mu_{1,\mathrm{ML}})^T\right] \\ &+ \frac{N_0}{N}\left[\frac{1}{N_0}\sum_{t_n=0}^{N}(x_n - \mu_{0,\mathrm{ML}})(x_n - \mu_{0,\mathrm{ML}})^T\right],\end{aligned}$$

which is a weighted linear combination of the estimated covariance matrices of the different groups.

Linear Discriminant Analysis with two classes: Classification

- For the prediction task on new data x we then have the fit $p(c_1|x) \approx \sigma({w^*}^T x + w_0^*)$ with:

$$\begin{aligned} w^* &:= \Sigma_{\mathrm{ML}}^{-1}(\mu_{1,\mathrm{ML}} - \mu_{0,\mathrm{ML}}), \\ w_0^* &:= -\tfrac{1}{2}\mu_{1,\mathrm{ML}}^T \Sigma_{\mathrm{ML}}^{-1} \mu_{1,\mathrm{ML}} \\ &\quad + \tfrac{1}{2}\mu_{0,\mathrm{ML}}^T \Sigma_{\mathrm{ML}}^{-1} \mu_{0,\mathrm{ML}} + \ln\left(\frac{q_{\mathrm{ML}}}{1-q_{\mathrm{ML}}}\right), \end{aligned}$$

  where σ is the logistic sigmoid function.
- We then assign x to class c_1 if $\sigma({w^*}^T x + w_0^*) \geqslant \frac{1}{2}$ and to class c_0 otherwise.
- Problems with LDA:
  - The Gaussian is sensitive to outliers.
  - Computing w^*, w_0^* out of the parameter estimates adds a lot of variance to the prediction.
  - Linearity (and/or handcrafted features) restricts the application possibilities.
  - Maximum likelihood estimates are prone to overfitting.

Example: Linear Discriminant Analysis with two classes

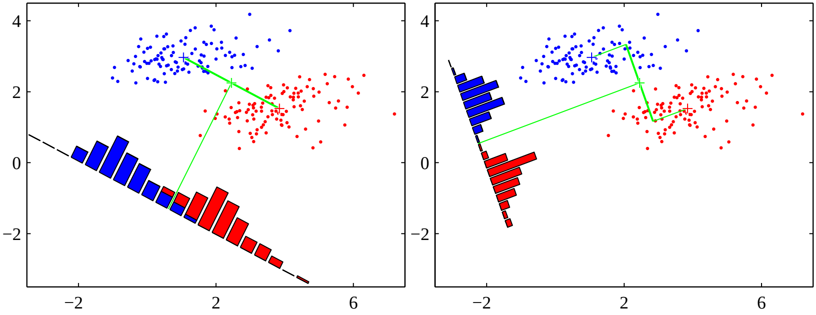

Figure: Left: Naive linear classifier projecting unto the line through the class means. Right: Linear classifier adjusting for the group covariances (Linear Discriminant Analysis). (Bishop 4.6)

Linear Discriminant Analysis (LDA) for Multiple Classes

- Given: Training set $D = (x_1, \dots, x_N)^T$ with targets $T = (t_1, \dots, t_N)^T$ of K classes $t_i \in \{c_1, \dots, c_K\}$.
- Prior: $p(c_k) =: q_k$, $k = 1, \dots, K$.
- LDA-assumption: $p(x|c_k) = \mathcal{N}(x|\mu_k, \Sigma)$ (same Σ for every k).
- (Unbiased) maximum likelihood estimates:

$$\begin{aligned} N_k &:= \#\{1 \leqslant n \leqslant N | t_n = c_k\}, \\ q_{k,\mathrm{ML}} &= \frac{N_k}{N}, \\ \mu_{k,\mathrm{ML}} &= \frac{1}{N_k} \sum_{n:t_n=c_k} x_n, \\ \tilde{\Sigma}_{\mathrm{ML}} &= \frac{1}{N-K} \sum_{k=1}^{K} \sum_{n:t_n=c_k} (x_n - \mu_{k,\mathrm{ML}})(x_n - \mu_{k,\mathrm{ML}})^T, \end{aligned}$$

- Posterior: $p(c_k|x) \approx \sigma_k(w_1^T x + w_{10}, \dots, w_K^T x + w_{K0})$ with:

$$w_j = \tilde{\Sigma}_{\mathrm{ML}}^{-1} \mu_{j,\mathrm{ML}}, \qquad w_{j0} = -\frac{1}{2} \mu_{j,\mathrm{ML}}^T \tilde{\Sigma}_{\mathrm{ML}}^{-1} \mu_{j,\mathrm{ML}} + \ln q_{j,\mathrm{ML}}.$$

- We assign x to class c_k if $\sigma_k > \sigma_j$ for all $j \neq k$, i.e.:
- Decision regions: $\mathcal{R}_k = \{x | w_k^T x + w_{k0} > w_j^T x + w_{j0}, \forall j \neq k\}$.
- Decision boundaries: $\mathcal{B}_{jk} = \{x | w_j^T x + w_{j0} = w_k^T x + w_{k0}\}$.
- For use of basis functions ϕ_m replace x with $\phi(x)$ everywhere.

Quadratic Discriminant Analysis (QDA) for Multiple Classes

- Given: Training set $D = (x_1, \dots, x_N)^T$ with targets $T = (t_1, \dots, t_N)^T$ of K classes $t_i \in \{c_1, \dots, c_K\}$.
- Prior: $p(c_k) =: q_k,\ k = 1, \dots, K$.
- QDA-assumption: $p(x|c_k) = \mathcal{N}(x|\mu_k, \Sigma_k)$.
- (Unbiased) maximum likelihood estimates:

$$\begin{aligned}
N_k &:= \#\{1 \leqslant n \leqslant N | t_n = c_k\}, \\
q_{k,\mathrm{ML}} &= \tfrac{N_k}{N}, \\
\mu_{k,\mathrm{ML}} &= \tfrac{1}{N_k} \textstyle\sum_{n:t_n=c_k} x_n, \\
\tilde{\Sigma}_{k,\mathrm{ML}} &= \tfrac{1}{N_k-1} \textstyle\sum_{n:t_n=c_k} (x_n - \mu_{k,\mathrm{ML}})(x_n - \mu_{k,\mathrm{ML}})^T,
\end{aligned}$$

- Posterior: $p(c_k|x) \approx \sigma_k(a_1(x), \dots, a_K(x))$ with:

$$a(x) = -\frac{1}{2}|\tilde{\Sigma}_{k,\mathrm{ML}}| - \frac{1}{2}(x-\mu_{k,\mathrm{ML}})^T \tilde{\Sigma}_{k,\mathrm{ML}}^{-1}(x-\mu_{k,\mathrm{ML}}) + \log q_{k,\mathrm{ML}}.$$

- We assign x to class c_k if $a_k(x) > a_j(x)$ for all $j \neq k$, i.e.:
- Decision regions: $\mathcal{R}_k = \{x | a_k(x) > a_j(x), \forall j \neq k\}$.
- Decision boundaries: $\mathcal{B}_{jk} = \{x | a_j(x) = a_k(x)\}$.
- For use of basis functions ϕ_m replace x with $\phi(x)$ everywhere.

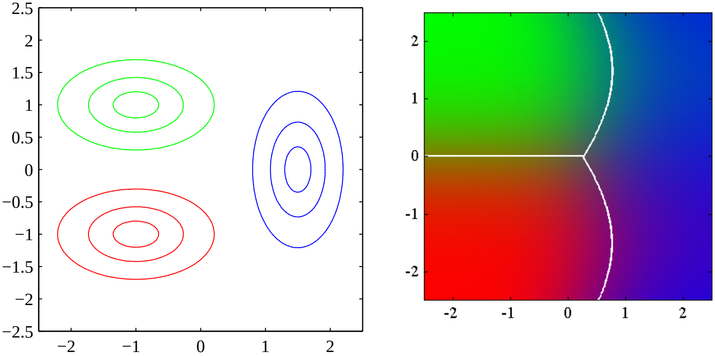

Figure: Left: Class conditional densities $p(x|c_k)$ for $K = 3$ classes and $x \in \mathbb{R}^2$. Right: Decision boundaries for Quadratic Discriminant Analysis (Bishop 4.11)

Example: The Use of Basis Functions

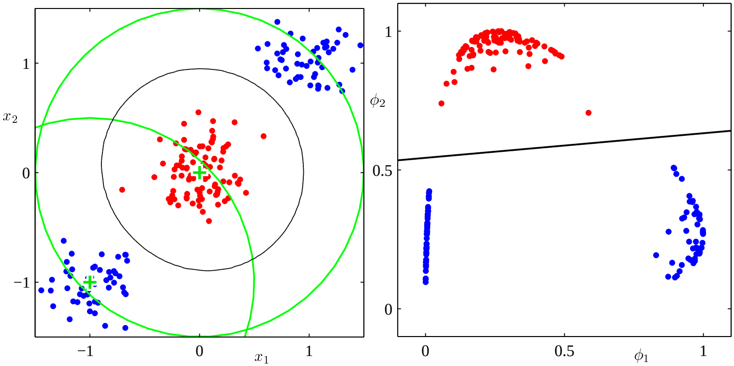

Figure: Left: Not linear separable data points $x = (x_{,1}, x_{,2}) \in \mathbb{R}^2$. Right: Using non-linear basis functions $\phi(x) = (\phi_1(x), \phi_2(x))$ makes the data linear separable. ϕ_i are the distances from the green crosses. The black linear decision boundary on the right corresponds to the non-linear decision boundary on the left. (Bishop 4.12)

Linear Classification: Logistic Regression for two classes (I)

- Given: Data set $D = (x_1, \ldots, x_N)^T$ with binary classes $T = (t_1, \ldots, t_N)^T$ with $t_i \in \{c_0, c_1\} = \{0, 1\}$.
- Basis functions: $\phi = \phi(x) = (\phi_0(x), \ldots, \phi_M(x))^T$ with $\phi_0 \equiv 1$.
- Model assumption of Logistic Regression: The posterior probability $p(c_1|\phi)$ is the sigmoid of a linear function in the feature vector ϕ:

$$p(c_1|\phi, w) = \sigma(w^T \phi)$$

  with weight vector $w = (w_0, \ldots, w_M) \in \mathbb{R}^{M+1}$.
- Conditional distribution:

$$\begin{aligned} p(t|\phi, w) &= \begin{cases} \sigma(w^T\phi) & \text{if} \quad t = 1, \\ 1 - \sigma(w^T\phi) & \text{if} \quad t = 0, \end{cases} \\ &= \sigma(w^T\phi)^t \cdot (1 - \sigma(w^T\phi))^{1-t}. \end{aligned}$$

- Conditional Likelihood (under i.i.d. assumptions):

$$
\begin{aligned}
p(T|\Phi, w) &= \prod_{n=1}^{N} p(t_n|\phi(x_n), w) \\
&= \prod_{n=1}^{N} \sigma(w^T\phi(x_n))^{t_n} \cdot (1-\sigma(w^T\phi(x_n)))^{1-t_n} \\
&= \prod_{n=1}^{N} y_n^{t_n} \cdot (1-y_n)^{1-t_n}
\end{aligned}
$$

  with $y_n := \sigma(w^T\phi(x_n))$. Put $Y = (y_1, \ldots, y_N)^T$.
- For the maximum likelihood approach we either needed to know $p(\Phi|w)$ or at least assume that it does not depend on w.
- This leads to maximizing the conditional likelihood w.r.t. w.
- This is equivalent to minimizing the cross-entropy error:

$$
\begin{aligned}
E(w) &= -\ln p(T|\Phi, w) \\
&= -\sum_{n=1}^{N} [t_n \ln y_n + (1-t_n)\ln(1-y_n)]
\end{aligned}
$$

- $E(w)$ is convex, but no closed form solution exists (due to the non-linearity of σ).
- For minimizing $E(w)$ use numerical methods, (stochastic) gradient descent or:

Iteratively Reweighted Least Squares (IRLS)

- Goal: Minimize a convex function $E(w)$.
- Iterated Reweighted Least Squares algorithm with Newton-Raphson update rule:
  - Carefully take a initialization $w^{(0)}$ (usually $w^{(0)} = 0$ works).
  - Iterate until only "small" changes occur:
  - Calculate the gradient $\nabla E(w)$ at $w = w^{(t)}$.
  - Calculate the Hessian matrix $H(w) = \nabla\nabla E(w)$ at $w = w^{(t)}$ and invert it.
  - Newton-Raphson update:

$$w^{(t+1)} := w^{(t)} - H(w^{(t)})^{-1} \nabla E(w^{(t)}).$$

- This will converge to the minimum of $E(w)$ (if existent, and not "overshooting").
- In these rare cases step-size-halving will ensure convergence.
- In the case $E(w)$ is the cross-entropy error, this update rule uses all data points of the training set at each step at once.
- In case $E(w)$ is the least-squares error for linear regression, this rule will computing the closed form solution (in one step).

Geometry of Gradient Descent and Newton Optimization

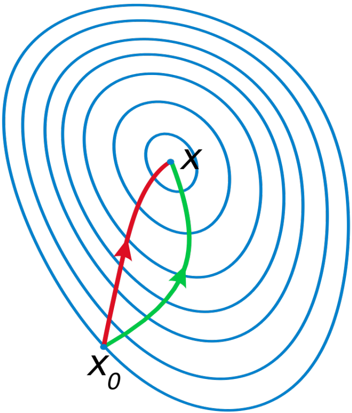

Figure: Contours of a convex error function $E(w)$. $w = X$ is the minimum of $E(w)$ and $w = X_0$ a starting point. Green: Gradient descent follows the steepest descent at each point, othogonal to the contours. Red: Newton-Raphson method also takes the curvature into account to shorten the way. (Source: Wikipedia - Newton's method in optimization)

Iteratively Reweighted Least Squares for Cross-Entropy

- Given: Data set $D = (x_1, \ldots, x_N)^T$ with binary classes $T = (t_1, \ldots, t_N)^T$ with $t_i \in \{c_0, c_1\} = \{0, 1\}$.
- Put $\Phi = (\phi(x_1), \ldots, \phi(x_N))^T$ with basis functions $\phi(x) = (\phi_0(x), \ldots, \phi_M(x))^T$.
- Put $y_n = \sigma(w^T \phi(x_n))$ and $Y = (y_1, \ldots, y_N)^T$ with $w \in \mathbb{R}^{M+1}$.
- $E(w) = -\sum_{n=1}^{N} \left[t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\right]$.
- Gradient: $\nabla E(w) = \sum_{n=1} (y_n - t_n)\phi(x_n) = \Phi^T(Y - T)$.
- Hessian: $H(w) = \sum_{n=1}^{N} y_n(1 - y_n)\phi(x_n)\phi(x_n)^T = \Phi^T R \Phi$, with
- diagonal matrix R with entries $R_{nn} = y_n(1 - y_n)$.
- Newton-Raphson update:

$$
\begin{aligned}
w^{(t+1)} &:= w^{(t)} - H(w^{(t)})^{-1} \nabla E(w^{(t)}) \\
&= w^{(t)} - (\Phi^T R^{(t)} \Phi)^{-1} \Phi^T (Y^{(t)} - T) \\
&= (\Phi^T R^{(t)} \Phi)^{-1} \left[\Phi^T R^{(t)} \Phi w^{(t)} - \Phi^T (Y^{(t)} - T)\right] \\
&= (\Phi^T R^{(t)} \Phi)^{-1} \Phi^T R^{(t)} Z^{(t)}, \\
Z^{(t)} &= \Phi w^{(t)} - (R^{(t)})^{-1} (Y^{(t)} - T)
\end{aligned}
$$

Logistic Regression for multiple classes

- Data $D = (x_1, \dots, x_N)^T$ with $T = (t_1, \dots, t_N)^T$ of K-dim one-vs-the-rest vectors $t_i = (0, \dots, 1, \dots, 0)^T$.
- Model assumption of Logistic Regression:

$$p(c_k | \phi, w_1, \dots, w_k) = \sigma_k(w_1^T \phi, \dots, w_K^T \phi),$$

with weight vectors $w_k = (w_{k,0}, \dots, w_{k,M}) \in \mathbb{R}^{M+1}$.
- Put $y_{nk} := \sigma_k(w_1^T \phi(x_n), \dots, w_K^T \phi(x_n))$.
- Conditional likelihood with $W = (w_1, \dots, w_K)^T$:

$$p(T | \Phi, W) = \prod_{n=1}^{N} \prod_{k=1}^{K} p(c_k | \phi(x_n), W)^{t_{nk}} = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}}.$$

- Minimize the cross-entropy error:

$$E(W) = -\ln p(T | \Phi, W) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln y_{nk}.$$

- Gradient: $\nabla_{w_j} E(W) = \sum_{n=1}^{N} (y_{nj} - t_{nj}) \phi(x_n)$
- Hessian: $\nabla_{w_k} \nabla_{w_j} E(W) = -\sum_{n=1}^{N} y_{nk} (\mathbb{1}_{nj} - y_{nj}) \phi(x_n) \phi(x_n)^T$.