Machine Learning 1

Lecture 05 - Linear Classification

Patrick Forré

- Given an input vector $x = (x_{,1}, \ldots, x_{,D}) \in \mathbb{R}^D$ we want to assign it to / predict one of the K classes $t \in \{c_1, \ldots, c_K\}$.
- The strategy will be to devide $\mathbb{R}^D$ into decision regions each assigned to a class and whose boundaries are called decision bounderies.

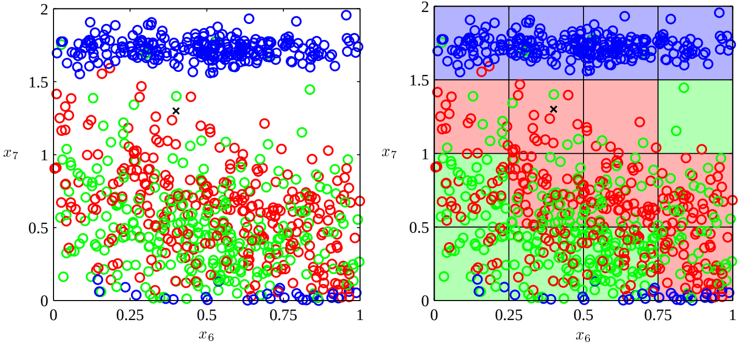


Figure: Classification via decision regions (Bishop 1.19 + 1.20)

Linear Classification

- Linear classification means that we consider linear $(D-1)$-dimensional hyperplanes as decision boundaries.
- Data sets whose classes can be separated exactly by linear decision surfaces are called linear separable.

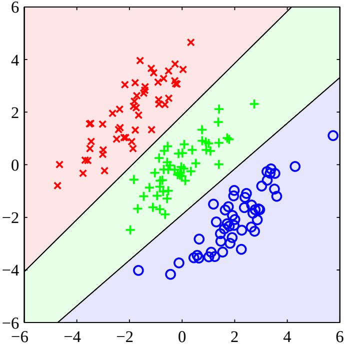


Figure: Linear separable data set (Bishop 4.5)

Multiple Classes: one-vs-the-rest dummies

- Situation: Predict one of the K classes $\{c_1, \dots, c_K\}$ of a random variable T with $K \geqslant 2$.
- For $j = 1, \dots, K$ define the one-vs-the-rest dummy variable:

$$\mathbb{1}_{c_j}(T) := \begin{cases} 1 & \text{if} \quad T = c_j \\ 0 & \text{if} \quad T \neq c_j. \end{cases}$$

- I.a.w. represent c_j as the vector $(0, \dots, 0, \overbrace{\quad 1 \quad}^{j\text{-th}}, 0, \dots, 0)^T$.
- Predicting the K-classed variable $T \in \{c_1, \dots, c_K\}$ is then equivalent to the K-fold binary prediction of $\mathbb{1}_{c_j}(T) \in \{0, 1\}$ for $j = 1, \dots, K$.
- So in most cases we can reduce to the case where T is a binary variable with classes $\{0, 1\}$. But not always:

Example: one-vs-the-rest failure

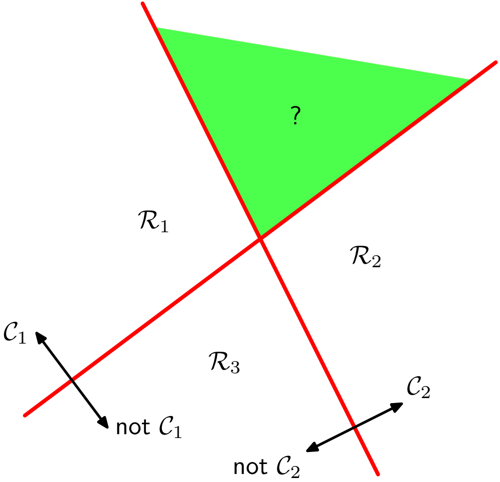


Figure: The one-vs-the-rest construction for $K \geqslant 3$ classes leading to ambiguous regions (green) (Bishop 4.2)

Classification: Three approaches

We will analyse three different approaches for the classification task:

1. Discriminant Functions: Learn functions $y_k(x, w_k)$ which will give equations for the decision boundaries associated to classes $\{c_1, \dots, c_K\}$.
   We will consider generalized linear discriminant functions of the form:
   $$y_k(x, w_k) = g(\sum_{m=0}^{M} w_{km}\phi_m(x)),$$
   where ϕ_m are "features" of x and g is a fixed (non-linear, monotonous) activation function. For simplicity we will assume $\phi_m(x) = x_m$.
2. Probabilistic Generative Models: Model the class-conditional densities $p(x|c_j)$ as well as the class priors $p(c_j)$, and then use Bayes' rule to compute the posterior density $p(c_j|x)$.
3. Probabilistic Discriminative Models: Maximize a likelihood function attached to the density $p(c_j|x)$.

Linear Discriminant Functions: Two Classes

- For D-dimensional input vector $x = (x_{,1}, \dots, x_{,D})^T \in \mathbb{R}^D$ and two classes $\{c_0, c_1\}$, in the simplest case, we consider real valued linear <u>linear discriminant functions</u>:

$$y(x, w) = w^T x + w_0,$$

where $w \in \mathbb{R}^D$ is called <u>weight vector</u> and $w_0 \in \mathbb{R}$ the <u>bias</u>.
- $\mathcal{B} = \{x \in \mathbb{R}^D | y(x, w) = 0\}$ is called the <u>decision boundary</u>.
- We then have the <u>decision regions</u> for x given by
$\mathcal{R}_0 = \{x \in \mathbb{R}^D | y(x, w) < 0\}$ (for class c_0) and
$\mathcal{R}_1 = \{x \in \mathbb{R}^D | y(x, w) > 0\}$ (for class c_1).
- The vector w stands orthogonal onto the decision boundary and points into the c_1-region:
If $y(x_A, w) = 0$ and $y(x_B, w) = 0$ then $w^T(x_A - x_B) = 0$.
If $y(x_C, w) > 0$ then $w^T(x_C - x_A) > 0$.
- w_0 determines the signed normal distance of the decision boundary from the origin $x_O = 0$: $\frac{w^T x_A}{||w||} = -\frac{w_0}{||w||}$.

Example: Geometry of Linear Discriminant Functions

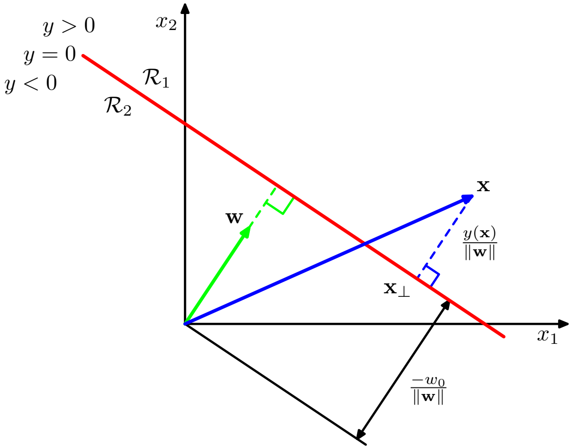


Figure: Decision surface $y(x) = 0$ in red is orthogonal to w. Signed normal distance of a point x to the decision surface is $y(x)/||w||$ in blue. (Bishop 4.1)

Linear Discriminant Functions: Multiple Classes

- For D-dimensional input vector $x = (x_{,1}, \ldots, x_{,D})^T \in \mathbb{R}^D$ and K classes $\{c_1, \ldots, c_K\}$, we now consider the K linear functions:

$$y_k(x) = w_k^T x + w_{k,0},$$

where every $w_k \in \mathbb{R}^D$ and $k = 1, \ldots, K$.

- The region for assigning an x to class c_k then is:

$$\mathcal{R}_k = \{x \in \mathbb{R}^D | y_k(x) > y_j(x) \forall j \neq k\}.$$

- The decision boundary $\mathcal{B}_{kj}$ between c_k and c_j is given by the $(D-1)$-dimensional hyperplane:

$$\begin{aligned} \mathcal{B}_{kj} &= \{x \in \mathbb{R}^D | y_k(x) = y_j(x)\} \\ &= \{x \in \mathbb{R}^D | (w_k - w_j)^T x + (w_{k,0} - w_{j,0}) = 0\}. \end{aligned}$$

- The regions $\mathcal{R}_k$ are convex and connected.

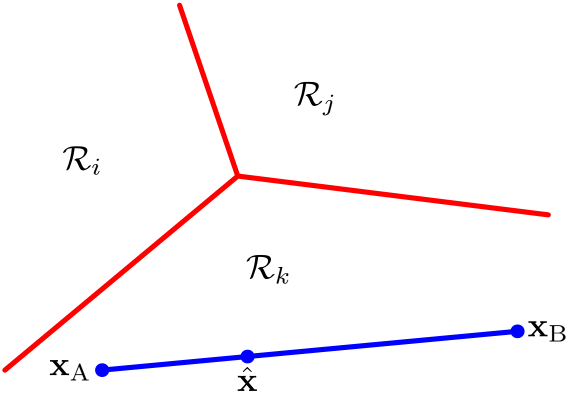


Figure: Decision regions for multiclass linear discriminant. Decision boundaries in red. The blue line illustrates the convexity and connectedness of the decision regions. (Bishop 4.3)

Linear Discriminant Functions: Short Notations

- For D-dimensional input vector $x = (x_{,1}, \ldots, x_{,D})^T \in \mathbb{R}^D$ and K classes $\{c_1, \ldots, c_K\}$, we have the K linear functions:

$$y_k(x) = w_{k,0} + w_k^T x.$$

- We put $\tilde{w}_k = (w_{k,0}, w_{k,1}, \ldots, w_{k,D})^T$ and $\tilde{x} = (1, x_{,1}, \ldots, x_{,D})^T$. Then:

$$y_k(x) = \tilde{w}_k^T \tilde{x}.$$

- Further define the $(D+1) \times K$-matrix $\tilde{W} = (\tilde{w}_1, \ldots, \tilde{w}_K)$ and $y(x) = y(x, \tilde{W}) = (y_1(x), \ldots, y_K(x))^T$.
- Then we get the brief notation of a vector valued function:

$$y(x) = \tilde{W}^T \tilde{x}.$$

- We will interpret the components of $y(x)$ as "probabilities" and assign an x to class c_k if $k = \operatorname{argmax}_{j=1,\ldots,K} y_j(x)$.

$$\mathcal{R}_k = \{x \in \mathbb{R}^D | \max_{j=1,\ldots,K} y_j(x) = y_k(x)\}.$$

Linear Regression for Classification: Sum-of-squares error

- Given the $N \times D$-data matrix $X = (x_1, \dots, x_N)^T$ with the $N \times K$-target matrix $T = (t_1, \dots, t_N)^T$, where every $t_i = (0, \dots, 1, \dots, 0)^T \in \{0, 1\}^K$ is given as a one-vs-the-rest vector, we define the $N \times (D+1)$-matrix
$$\tilde{X} = (\tilde{x}_1, \dots, \tilde{x}_N)^T.$$
- The sum-of-squares error function can conveniently written as:
$$\begin{aligned} E(X, \tilde{W}) &= \tfrac{1}{2} \operatorname{Tr}[(\tilde{X}\tilde{W} - T)^T(\tilde{X}\tilde{W} - T)] \\ &= \tfrac{1}{2} \sum_{k=1}^{K} \sum_{n=1}^{N} \left(\sum_{d=0}^{D} x_{nd} w_{dk} - t_{nk}\right)^2. \end{aligned}$$
- The least-squares minimizer then is:
$$\tilde{W}_{\mathrm{LS}} = (\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T T.$$
- The learned function then is:
$$y_{\mathrm{LS}}(x) = y(x, \tilde{W}_{\mathrm{LS}}) = \tilde{W}_{\mathrm{LS}}^T \tilde{x} = T^T \tilde{X}(\tilde{X}^T\tilde{X})^{-1}\tilde{x}.$$
- We assign x to class c_k for $k = \operatorname{argmax}_{j=1,\dots,K} y_{\mathrm{LS},j}(x)$.

Problems: Sum-of-squares error function

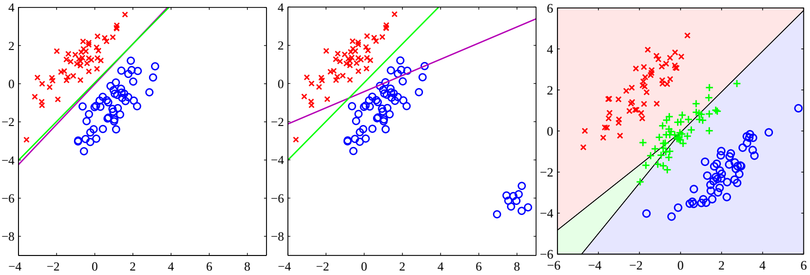


Figure: Least-square decision boundary (magenta) too sensitive to outliers and leading to too small regions (right). (Bishop 4.4 + 4.5)

Problems: Linear Regression for Classification

- Least-square decision boundary are too sensitive to outliers.
- For $K > 2$ some decision regions may become too small or are even ignored/masked.
- The components of y_{LS} are not interpretable as conditional probabilities. They may become negative or bigger than 1.

Linear Classification: Perceptron

- Let data $X = (x_1, \ldots, x_N)^T$ be given with target variables $t_i \in \{-1, 1\}$.
- Take $y(x, w) = g(w^T \phi(x))$ as functions with w including a bias and $\phi_0 \equiv 1$ and with activation function:

$$g(a) := \operatorname{sign}(a) := \begin{cases} 1 & \text{if} \quad a \geqslant 0 \\ -1 & \text{if} \quad a < 0. \end{cases}$$

- Perceptron criterion: Assign x to class c_1 if $w^T \phi(x) \geqslant 0$ (and c_{-1} if $w^T \phi(x) < 0$).
- For correct classification we need to find w such that for all (x, t) we have $w^T \phi(x) t > 0$.
- Perceptron error: $E_P(w) = -\sum_{n \in \mathcal{M}} w^T \phi(x_n) t_n$, where $\mathcal{M}$ is the set of all misclassified point.
- Minimization: Stochastic gradient descent with learning rate η, randomly chosen (x_n, t_n): $w^{(n+1)} = w^{(n)} + \eta \phi(x_n) t_n$.
- Theorem: If X is linear separable then the algorithm converges.

Example: Perceptron Algorithm

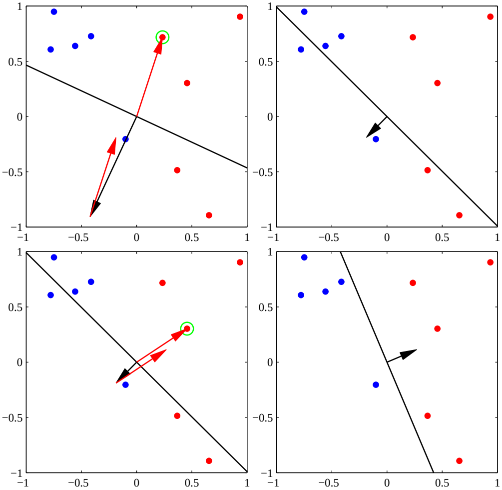
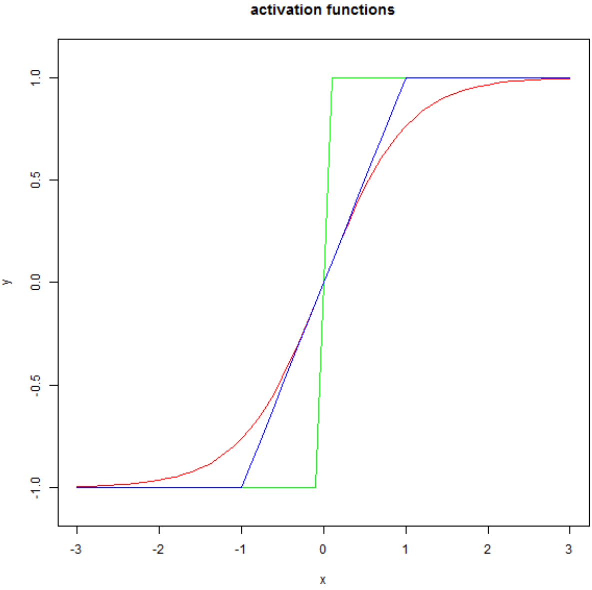


Figure: Perceptron algorithm in 2-dim feature space. Decision boundary in black. w as black arrow pointing to red side. Misclassified point in green. Update of w by adding red vector. (Bishop 4.7)

Problems: Perceptron

- Perceptron only works with two classes.
- There might be many solutions depending of initialization and presentation order of data.
- If data set is not linear separable the perceptron does not converge.
- Based on linear combination of fixed basis functions.

Other Activation Functions



Misclassification Rate (I)

- Assume we have input vectors $x \in \mathbb{R}^D$ together with one of the K classes $t \in \{c_1, \ldots, c_K\}$.
- We devide $\mathbb{R}^D$ into decision regions $\mathcal{R}_i$, $i = 1, \ldots, K$.
- Every observation x will then have a true class c_k and an assigned class $\mathcal{R}_j$.
- Counting all instances of N observations $x_1, \ldots, x_N$ we get a $K \times K$-matrix, the <u>confusion matrix</u>:

$$\begin{array}{c} \\ c_1 \\ c_2 \\ \vdots \\ c_K \end{array} \begin{array}{c} \begin{array}{cccc} \mathcal{R}_1 & \mathcal{R}_2 & \cdots & \mathcal{R}_K \end{array} \\ \begin{pmatrix} 6 & 1 & 1 & 1 \\ 5 & 3 & 0 & 1 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 7 \end{pmatrix} \end{array} = C$$

- The diagonal elements count the correct classified ones.
- The off-diagonal elements count the falsely classified ones.

Misclassification Rate (II)

- Now assume that the observations are drawn from a joint distribution $p(x, t)$, where t is the class of x.
- The misclassification rate/error is then:

$$\begin{aligned} p(\text{mistake}) &= \sum_{i \neq j} p(x \in \mathcal{R}_i, c_j) \\ &= \sum_{i \neq j} \int_{\mathcal{R}_i} p(x, c_j) dx, \\ &= 1 - \sum_{k=1}^{K} \int_{\mathcal{R}_k} p(x, c_k) dx. \end{aligned}$$

- Strategy: Create the decision regions $\mathcal{R}_i$ such that the misclassication error is minimal.
- Minimizing the misclassication rate leads to the rule:
- Assign x to class c_k if $p(x, c_k) > p(x, c_j)$ for all $j \neq k$.
- Equivalently: if the posterior $p(c_k|x) > p(c_j|x)$ for all $j \neq k$.

Example: Misclassification Rate

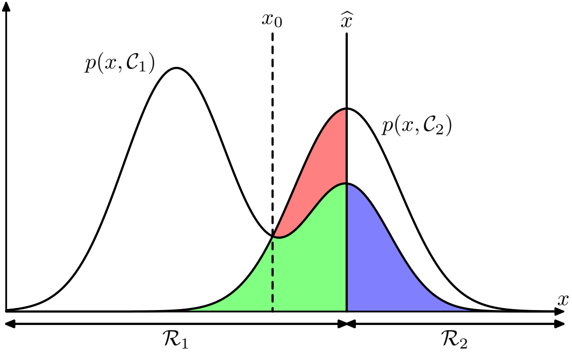


Figure: Distribution for two classes. Misclassification happens in coloured area: red + green: points from class c_2 labeled as class c_1; blue: points from class c_1 labeled as c_2. Decision boundary at $\hat{x}$. Minimal misclassification error for $\hat{x} = x_0$. (Bishop 1.24)

Problems: Minimizing the Misclassification Rate

- Error types might have a different impact:
- Example: Labeling a healthy person as cancerous is annoying. But labeling a person with cancer as healthy has serious consequences.
- Furthermore, if cancer occurs only in 1% of all cases, labeling everyone as healthy gives a misclassification rate as low as 1%.

Expected Loss

- Possible solution: Weighting different error types differently. Use of a loss matrix:

$$\begin{array}{c} \\ \text{true cancer} \\ \text{true normal} \end{array} \begin{array}{c} \begin{array}{cc} \text{label cancer} & \text{label normal} \end{array} \\ \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} \end{array} = L$$

- The expected loss is:

$$\mathbb{E}[\text{Loss}] = \sum_{i \neq j} L_{ji} \cdot \int_{\mathcal{R}_i} p(x, c_j) dx.$$

- Strategy: Create the decision regions $\mathcal{R}_i$ such that the expected loss is minimal.
- Minimizing the expected loss leads to the rule:
- Assign x to the label c_k for which the weighted posterior $\sum_{j=1}^{K} L_{jk} \cdot p(c_j|x)$ is minimal.

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 $\{c_1, \ldots, c_K\}$ we can write the posterior as:

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

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

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

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

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

Sigmoid function

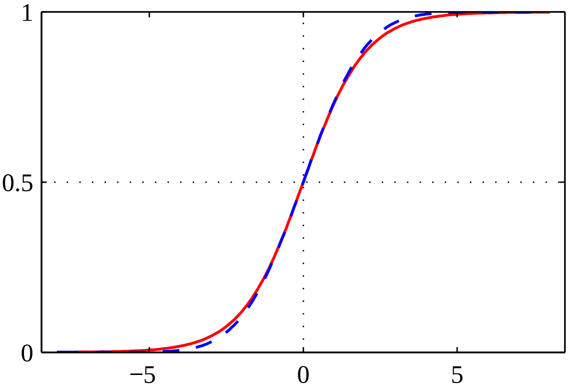


Figure: Sigmoid function $\sigma(a) = \frac{1}{1+\exp(-a)}$ (in red) and scaled cumulative normal distribution $\Phi(a) = \int_{-\infty}^{a} \mathcal{N}(x|0,1)dx$ (in blue). We have the symmetry property $\sigma(-a) = 1 - \sigma(a)$ and derivative $\sigma'(a) = \sigma(a)(1 - \sigma(a))$. (Bishop 4.9)