EE-411: Fundamentals of inference and learning **2022-2023**

Lecture 5: (Generalized) Linear models: Ridge, Lasso, Logistic & Hinge

Lecturer: Florent Krzakala

Given a sample $\{\vec{x_i}, y_i\}_{i=1,..,n}$ where $\vec{x}_i \in \mathbb{R}^d$ are independent variables and $y_i \in \mathbb{R}$ labels related to them. We are looking for a function $f(\cdot)$ that, given some new data $\vec{x}$, is able to predict the respective label y. The simplest model is the linear model defined by:

$$f(\vec{x}) = \alpha + \vec{\beta}\vec{x} \tag{5.1}$$

We can simplify this further by expressing (5.1) through a simple scalar product:

$$\vec{w} = \begin{bmatrix} \vec{\beta} \\ \alpha \end{bmatrix} \in \mathbb{R}^{d+1} \qquad \vec{u} = \begin{bmatrix} \vec{x} \\ 1 \end{bmatrix} \in \mathbb{R}^{d+1}$$

$$f(\vec{x}) = \vec{w} \cdot \vec{u} \tag{5.2}$$

R We will write $f_{\vec{w}}(\vec{x}) = \vec{w} \cdot \vec{x}$ where we assume that there is a 1 in $\vec{x}$ to take care of the constant (intercept).

5.1 Regression

Given 5.2, we estimate the parameters of $\vec{w}$ by minimizing the quadratic loss function:

$$\sum_i (y_i - f(\vec{x}_i))^2 \tag{5.3}$$

5.1.1 Ridge Regression

The Ridge regression is a method to solve the minimization problem of the equation (5.3). The ridge problem is the following:

$$\vec{w}^*_{Ridge} = \operatorname{argmin} \sum_i (y_i - \vec{w} \cdot \vec{x}_i)^2 + \lambda \left\|\vec{w}\right\|_2^2 \tag{5.4}$$

$$= \operatorname{argmin} \left[\left\|\vec{y} - X\vec{w}\right\|_2^2 + \lambda \left\|\vec{w}\right\|_2^2\right] \tag{5.5}$$

Where $\vec{y} \in \mathbb{R}^n$, $\vec{w} \in \mathbb{R}^p$ and X an $n \times p$ matrix.

R The $\mathrm{l_p}$-norm is by definition: $\left\|\vec{v}\right\|_p = \left(\sum_i |v_i|^p\right)^{1/p}$

5.1.1.1 Scalar case

We can find w^* analytically in the scalar case. To that end, we should minimize: $(y - xw)^2 + \lambda w^2$. If we derive this expression with respect to w, we get:

$$\begin{aligned}\frac{\partial}{\partial w}\left[(y-xw)^2+\lambda w^2\right] &= -x(y-xw)^2+\lambda w^* = 0\\ &\Leftrightarrow (x^2+\lambda)w^* = xy\end{aligned}$$

5.1.1.2 Vector case

The solution to the equation (5.5) is given by :

$$-X^T(\vec{y} - X\vec{w}) + \lambda\vec{w} = \vec{0}$$

Which gives :

$$(X^TX + \lambda\mathbb{1})\vec{w}^* = X^T\vec{y} \qquad Normal\ equation \tag{5.6}$$

We can take a look at some interesting properties :

SVD : Singular value decomposition

$$X = U\Sigma V$$

- X is of size $(n \times d)$
- U is of size $(n \times n)$ and is *orthogonal* $(U^TU = \mathbb{1})$
- Σ is of size $(n \times d)$ and is filled with zeros except on the diagonal that contains the singular values σ_i

$$\begin{bmatrix}\sigma_1 & 0 & \cdots & 0\\ 0 & \sigma_2 & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0\\ \vdots & \ddots & \ddots & \sigma_d\\ 0 & \cdots & \cdots & 0\end{bmatrix} \text{when } d < n \qquad \text{or} \qquad \begin{bmatrix}\sigma_1 & 0 & \cdots & \cdots & 0\\ 0 & \sigma_2 & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ 0 & \cdots & 0 & \sigma_n & 0\end{bmatrix} \text{when } d > n$$

- V is of size $(d \times d)$ and is *orthogonal* $(V^TV = \mathbb{1})$

If we apply the SVD reformulation to X^TX we get:

$$\begin{aligned}X^TX &= V^T\Sigma^TU^TU\Sigma V\\ &= V^T\Sigma^2V\\ &= V^T\begin{bmatrix}\sigma_1^2 & 0 & \cdots & 0\\ 0 & \sigma_2^2 & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0\\ 0 & \cdots & 0 & \sigma_n^2\end{bmatrix}V\end{aligned}$$

So X^TX is *positive semi-definite.* This means that all the eigenvalues are ≥ 0.

$X^TX + \lambda\mathbb{1}$ is *positive definite*. This is because λ makes sure that none of the eigenvalues are 0.

The number of singular values is $\min(n, d)$

We can define the optimal w for RIDGE in 2 ways depending on the size of X:

$$\vec{w}^*_{RIDGE} = \frac{1}{X^TX + \lambda\mathbb{1}_{d\times d}} X^T\vec{y} \qquad \text{used when } d < n$$

Where:

- $A = X^TX$ is the *auto-correlation* matrix $(d \times d)$
- A_{ij} tells how feature i is correlated with feature j

$$\vec{w}^*_{RIDGE} = X^T \frac{1}{XX^T + \lambda\mathbb{1}_{n\times n}} \vec{y} \qquad \text{used when } n < d$$

Where:

- $K = XX^T$ is the *Gram* matrix $(n \times n)$
- K_{ij} is the inner product of sample i with sample j : $\vec{x}_i \cdot \vec{x}_j$

R Depending on the value of n and d, either one of the formulas should be used. But if n and d are big, then we are not in a good position. There are solutions to this situation that will be discussed below.

5.1.2 Ordinary Least Square (OLS)

The Ordinary Least Square problem consists in minimizing the squared euclidean distance between a vector $\hat{y}$ and its approximation calculated by multiplying the input X with a weight vector $\vec{w}$.

$$\vec{w}^*_{OLS} = \text{argmin } \|\vec{y} - X\vec{w}\|_2^2$$

If $n \geq d$: The optimisation problem is well defined and there exists a unique solution: $\vec{w}^*_{OLS} = (X^TX)^{-1}X^TY$

If $n < d$: There are infinitely many solutions. One of them is $\vec{w}^*_{LN} = X^T(XX^T)^{-1}Y$.

Obtaining infinitely many solutions is a real issue which can be erased by adding a regularization term.

5.1.3 LASSO (L1-penalty)

The LASSO problem L1-regularized variant for the OLS problem:

$$\vec{w}^*_{LASSO} = \text{argmin } \frac{1}{2}\|\vec{y} - X\vec{w}\|_2^2 + \lambda\|\vec{w}\|_1$$

where λ is the regularization parameter and $\|.\|_1$ is the L1-norm defined as $\sum_i |w_i|$.

Lasso brings sparsity to the solution and is therefore interesting for interpretability as it selects the most relevant features and tends to set the other coefficients to zero. The higher λ, the sparser the solution becomes.

5.1.4 Parenthesis on Linear Algebra

5.1.4.1 Singular value Decomposition

Let us define the samples $\vec{x_1}, \vec{x_2}, \ldots, \vec{x_n} \in \mathbb{R}^d$ and the matrix X of size $n \times d$ such that :

$$X = \begin{bmatrix} \longleftarrow & \vec{x_1} & \longrightarrow \\ & \vdots & \\ \longleftarrow & \vec{x_n} & \longrightarrow \end{bmatrix}$$

Theorem 5.1 *Any matrix can be decomposed as : $X = U\Sigma V$. This is called the SVD ("Singular Value Decomposition).*

- *U: orthonormal matrix of size $n \times n$ with $UU^T = \mathbb{1}$*
- *V: orthonormal matrix of size $d \times d$ with $VV^T = \mathbb{1}$*
- *Σ: Values on the diagonal are called singular values*
- *U, V can be viewed as pure rotations*

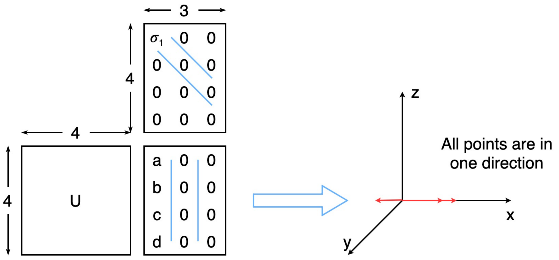
R The singular value decomposition *always exists* ! There are no conditions on the matrix X.

5.1.4.2 Rank of a matrix

Definition 5.2 *The rank of a matrix is the number of non-zero singular values. This number is always smaller than* $\min(n, d)$*. The rank is the measure of the dimension of the space defined by X.*

Example

We can express $X = U\Sigma V$ differently with $XV^T = U\Sigma$. Both matrices are of size $n \times d$ and represent the same vectors as before but with a slight rotation applied to them.
We consider X of size 4×3 with only $\sigma_1 \neq 0$.
The resulting vectors are all contained in a single line.



- Rank 1: All vectors are in one direction ⇒ they occupy a single line.
- Rank 2: $\sigma_1 > 0, \sigma_2 > 0$. All vectors occupy a single plane. Multiplying with V aligns the plane of X with the axis.
- Rank 3: $\sigma_1 > 0, \sigma_2 > 0, \sigma_3 > 0$. All vectors occupy the whole $\mathbb{R}^3$ space.

Definition 5.3 *The rank of a matrix is the dimension of the span of the vectors. It can also be expressed as the "range of the vectors".*

Definition 5.4 $K = XX^T$ *is the* Gram Matrix. $\Rightarrow$ $\text{Rank}_K \leq \min(n, d)$
To be able to invert K, the number of singular values needs to be at least n.

Definition 5.5 $A = X^T X$ *is the* Auto-correlation Matrix. $\Rightarrow$ $\text{Rank}_A \leq \min(n, d)$
To be able to invert A, the number of singular values needs to be at least d.

R For numerical applications, one should always worry about the order of magnitude of singular values. If some are too close to 0, one should not invert the matrix. In short it is dangerous to invert a matrix with small eigenvalues.

5.1.5 LASSO VS RIDGE

As shown in figure 5.1 below, the evolution of the weights with respect to regularization parameter is piecewise linear in the L1 case unlike ridge regression in figure 5.2. This example has been taken from a regression task carried out on a Diabete Dataset with 10 features.

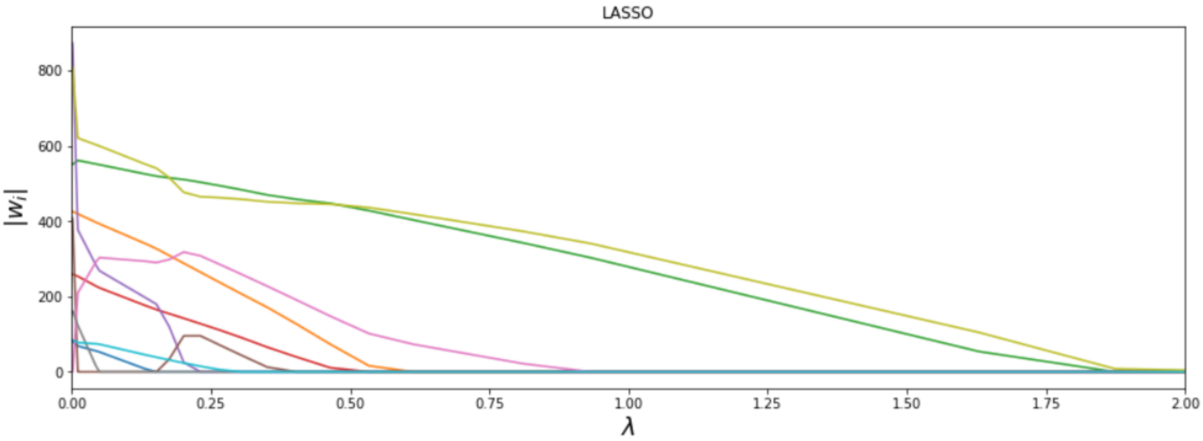


Figure 5.1: Weights with respect to λ for Lasso

1) Why more zeros with L1 than with L2?

The L1-regularizer penalizes very small values more than L2. This can be illustrated by first taking the definition of the p-norm:

$$\|x\|_p = \Big(\sum_i |x_i|^p\Big)^{1/p}$$

The graph shown in figure 5.3, shows the penalty applied to weights. In fact, the ideal would be to obtain very sparse weights as we would obtain by using a p-norm where p is smaller than 1. Indeed, this would result in a large penalty already on very small coefficients. However, any p-norm with $p < 1$ in a non-convex function which make the optimization problem much harder to solve, this is the reason why the L1-norm is used in practice when sparsity is desirable. On the contrary, the L2 regularizer tends to penalize large coefficients much more.

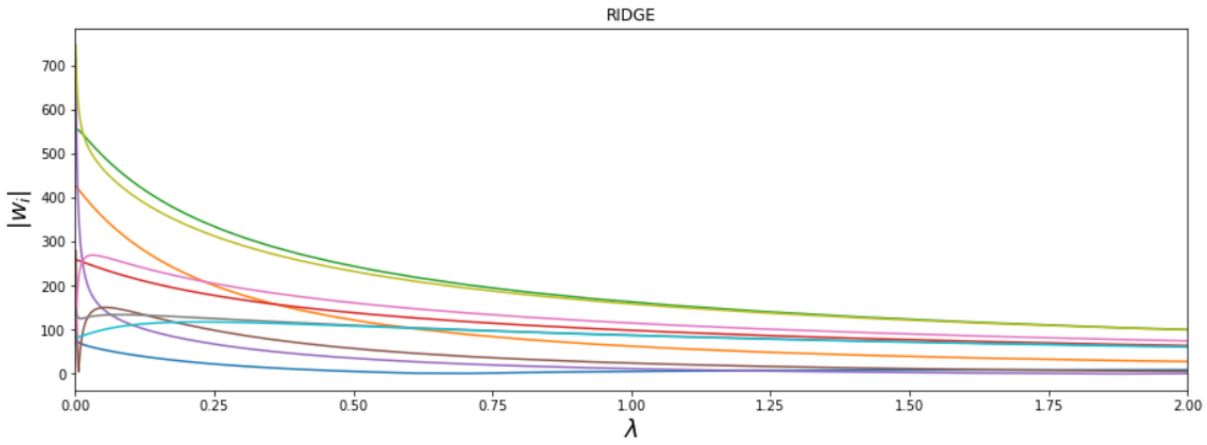


Figure 5.2: Weights with respect to λ for Ridge

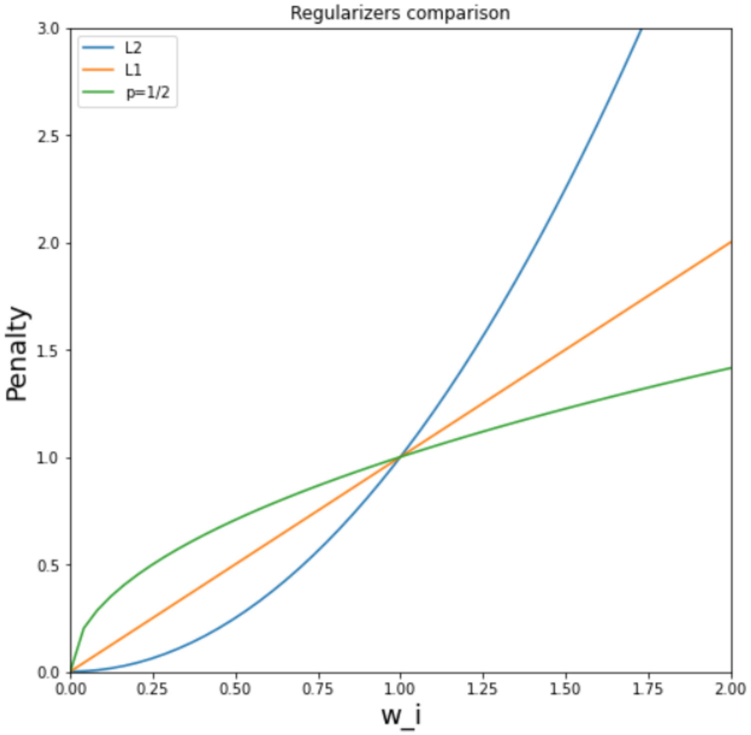


Figure 5.3: Comparison of $\|w_i\|_2$, $\|w_i\|_1$ and $\|w_i\|_{1/2}$

We can show that the p-norm is non-convex for $p < 1$. A necessary condition for a function to be convex is the non-negativity of its second derivative:

$$\frac{\partial}{\partial x}|x|^p = p|x|^{p-1}\text{sign}(x)$$
$$\frac{\partial^2}{\partial x^2}|x|^p = (p-1)p|x|^{p-2} \geqslant 0 \Leftrightarrow p \geqslant 1$$

2) Why L1 penalty is piecewise linear?

The optimality condition for the initial problem is

$$\frac{\partial}{\partial w}\frac{1}{2}\|\vec{y}-X\omega(\lambda)\|_2^2+\lambda\|\omega(\lambda)\|_1 \quad = \quad -X^\top\left(\vec{y}-Xw^*(\lambda)\right)+\lambda\,\mathrm{sign}\left(\omega^*(\lambda)\right)=0$$

Subsequently, we take the partial derivative with respect to λ for a particular interval $[\lambda_1, \lambda_2]$ on which $\mathrm{sign}(w^*(\lambda))$ does not change.

$$\begin{aligned}\frac{\partial}{\partial\lambda}X^\top\left(Xw^*(\lambda)-\vec{y}\right)+\lambda\,\mathrm{sign}\left(\omega^*(\lambda)\right) \quad &= \quad X^TX\dot{w}^*(\lambda)+\mathrm{sign}(w^*(\lambda))=0\\ &\implies \dot{w}^*(\lambda)=-\frac{\mathrm{sign}(w^*(\lambda))}{X^TX}\end{aligned}$$

Since $\mathrm{sign}(w^*(\lambda))$ is constant for the chosen interval, the latter equation expresses the constant slope of the function $w^*(\lambda)$ which is consequently linear. This explains why the weights regularized by the L1 norm are piecewise linear with respect to the regularization parameter λ.

Lasso is limited in certain cases where there is a group of correlated variables. Indeed this regularizer tends to keep only one non-zero value and ignore all other variables. In order to overcome this problem, the Elastic Net is introduced. The latter is simply the combination of a Lasso with a Ridge regularization.

5.1.6 Elastic Net

The Elastic Net is a simple combination of a L1 and a L2 regularization, resulting in the following optimization problem:

$$\vec{w}^*_{Elastic} \;=\; \mathrm{argmin}\;\|\vec{y}-X\vec{w}\|_2^2\,+\,\lambda_1\|w\|_1\,+\,\lambda_2\|w\|_2^2$$

The Elastic Net tends to promote sparse solutions while also penalizing large coefficient for the weight vector $\vec{w}$.

5.2 Classification

In a classification problem, there are some feature vectors $\vec{x_i}$ associated with certain labels y_i which can take values in $\{0,1\}$. The labels can also be written $s_i = \pm 1 = 2y_i - 1$.

The aim of a classification problem is to make predictions about labels given the associated feature vectors. To this aim, the following optimization problem tries to minimize the error made on predictions. In this framework, both Ridge or Lasso regularization can be used.

$$\vec{w}^* \;=\; \mathrm{argmin}\|\vec{s}-X\vec{w}\|_2^2+\mathit{penalty}$$

An alternative is to use the sign-loss function. The problem becomes

$$\vec{w}^* \;=\; \mathrm{argmin} \sum_{\mu=1}^{n} \frac{1 - \mathrm{sign}(\vec{x}_\mu \cdot \vec{w}\; s_\mu)}{2}$$

However, this objective is highly non-convex. This is the reason why, in practice, other types of regularization are preferred.

We can use convex surrogates like the exponential loss, the logistic loss or the Hinge loss instead. They are represented on the figure 5.4.

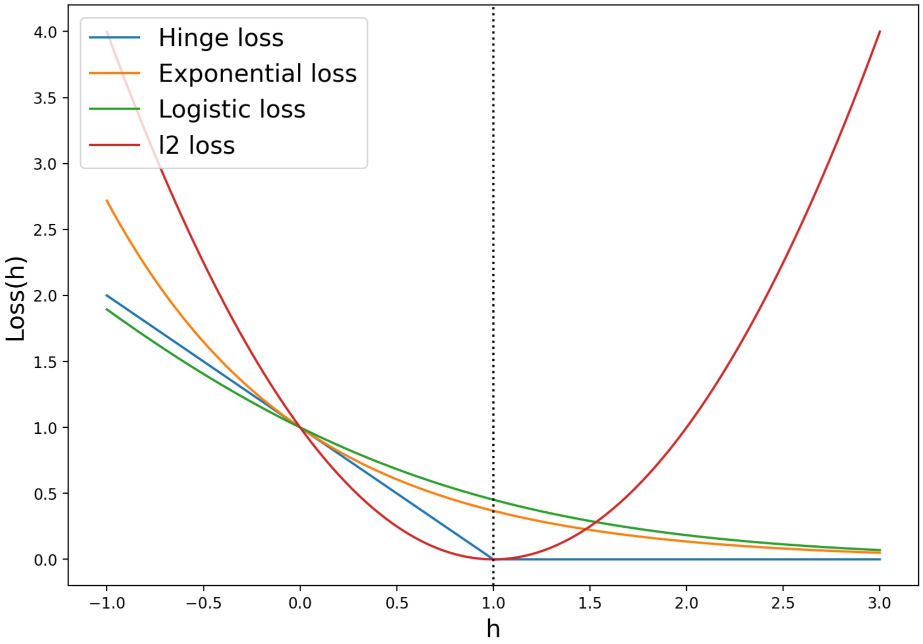


Figure 5.4: Convex losses

The problem of a logistic classification (or the cross-entropy) is:

$$\vec{w}^* = \mathrm{argmin} \sum_{\mu=1}^{n} \log_2\left(1 + e^{-s_\mu \vec{w}\cdot\vec{x}_\mu}\right) + \lambda_2 \left\|\vec{w}\right\|_2^2 + \lambda_1 \left\|\vec{w}\right\|_1$$

The Hinge problem (or the Support Vector Machine, SVM, problem) is:

$$\vec{w}^* = \mathrm{argmin} \sum_{\mu=1}^{n} \max\left[0, 1 - s_\mu \vec{w}\cdot\vec{x}_\mu\right] + \lambda \left\|\vec{w}\right\|_2^2$$

The logistic (or cross-entropy) loss is interesting to predict the probability that $\vec{x}_\mu$ has a label s_μ. Assume:

$$\begin{aligned}
\mathrm{P}(\vec{x}_\mu \text{ has label } s_\mu) &= \frac{e^{-s_\mu \vec{w}\cdot\vec{x}_\mu/2}}{e^{-s_\mu \vec{w}\cdot\vec{x}_\mu/2} + e^{s_\mu \vec{w}\cdot\vec{x}_\mu/2}} \\
&= \frac{e^{-h/2}}{e^{-h/2} + e^{h/2}} \\
&= \frac{1}{1+e^{-h}} \\
&= \sigma(h)
\end{aligned}$$

Where $\sigma(h)$ is the sigmoïd function represented in figure 5.5.

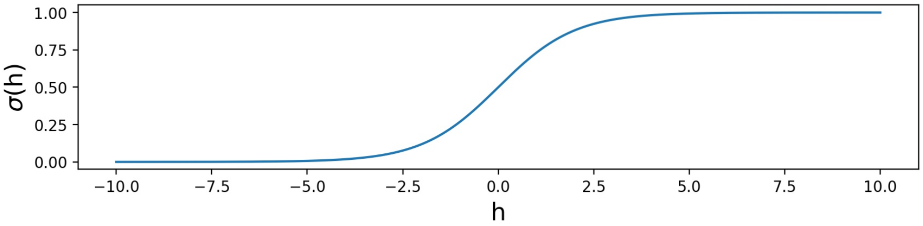


Figure 5.5: Sigmoïd function

This probability could be analysed using the the Kullback-Leibler divergence, D_{KL}, which is a measure of how one probability distribution is different from another.

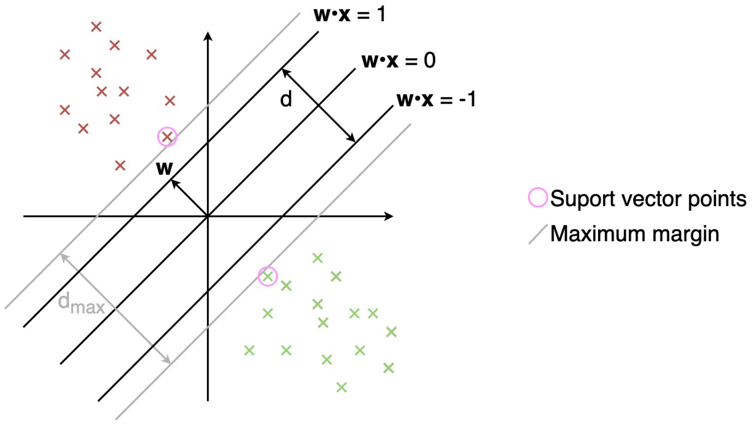
$$D_{\mathrm{KL}}(p\|q) = \sum_i \mathrm{p}_i \log_2 \frac{\mathrm{p}_i}{\mathrm{q}_i} \tag{5.7}$$

We can minimize the difference between the probability distribution of the logistic loss function and the true probability distribution.

$$\begin{aligned}
&\text{minimize } D_{\mathrm{KL}}(\mathrm{P}_{\mathrm{true}}\|\mathrm{P}_{\mathrm{logistic}}) \\
&\text{minimize } \sum_{\mu=1}^{n} \sum_{s_\mu=\pm1} \mathrm{P}_{\mathrm{true}}(s_\mu = s)\cdot \log_2 \frac{\mathrm{P}_{\mathrm{true}}(s_\mu = s)}{\mathrm{P}_{\mathrm{logistic}}(s_\mu = s)} \\
&\qquad\qquad = -\sum_{\mu=1}^{n} \sum_{s_\mu=\pm1} \mathrm{P}_{\mathrm{true}}(s = s_\mu)\cdot \log_2 \mathrm{P}_{\mathrm{logistic}}(s_\mu = s) \\
&\qquad\qquad = \sum_{\mu=1}^{n} \log_2\left(1+e^{-h}\right)
\end{aligned}$$

The Hinge loss is interesting because, when the data are linearly separable, it always finds a hyperplane that maximizes the distance between points with different labels. It defines a line in 2D, a plane in 3D or a hyperplane in bigger dimension.

The margin is the difference between lines (in 2D) that nearly touches the points of each class. This is given by equation (5.8).



$$\frac{d}{2} \cdot \frac{\vec{w}}{\|\vec{w}\|_2} \cdot \vec{w} = 1 \quad \Leftrightarrow \quad d = \frac{2}{\|\vec{w}\|_2} \tag{5.8}$$

The points that are touching the lines are called *support vector points*.