Machine Learning Lecture Notes

Predrag Radivojac

February 12, 2015

Given a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ the objective is to learn the relationship between features and the target. We usually start by hypothesizing the functional form of this relationship. For example,

$$f(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

where $\mathbf{w} = (w_0, w_1, w_2)$ is a set of parameters that need to be determined (learned) and $\mathbf{x} = (x_1, x_2)$. Alternatively, we may hypothesize that $f(\mathbf{x}) = \alpha + \beta x_1 x_2$, where $\boldsymbol{\theta} = (\alpha, \beta)$ is another set of parameters to be learned. In the former case, the target function is modeled as a *linear combination* of features and parameters, i.e.

$$f(\mathbf{x}) = \sum_{j=0}^{k} w_j x_j,$$

where we extended $\mathbf{x}$ to $(x_0 = 1, x_1, x_2, \ldots, x_k)$. Finding the best parameters $\mathbf{w}$ is then referred to as *linear regression problem*, whereas all other types of relationship between the features and the target fall into a category of *nonlinear regression*. In either situation, the regression problem can be presented as a probabilistic modeling approach that reduces to parameter estimation; i.e. to an optimization problem with the goal of maximizing or minimizing some performance criterion between target values $\{y_i\}_{i=1}^n$ and predictions $\{f(\mathbf{x}_i)\}_{i=1}^n$. We can think of a particular optimization algorithm as the *learning* or *training algorithm*.

1 Ordinary Least-Squares (OLS) Regression

One type of performance measure that is frequently used to find $\mathbf{w}$ is the sum of squared errors

$$\begin{aligned} E(\mathbf{w}) &= \sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2 \\ &= \sum_{i=1}^{n} e_i^2 \end{aligned}$$

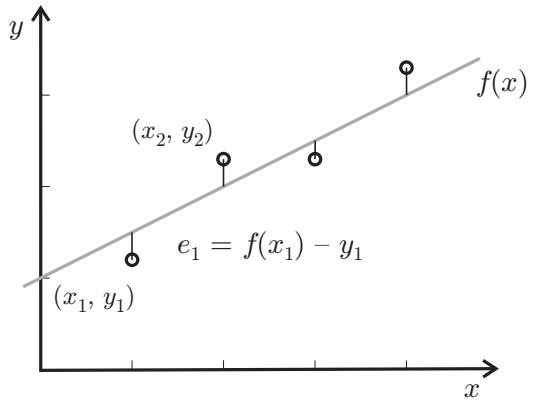

Figure 1: An example of a linear regression fitting on data set $\mathcal{D} = \{(1, 1.2), (2, 2.3), (3, 2.3), (4, 3.3)\}$. The task of the optimization process is to find the best linear function $f(x) = w_0 + w_1 x$ so that the sum of squared errors $e_1^2 + e_2^2 + e_3^2 + e_4^2$ is minimized.

which, geometrically, is the square of the Euclidean distance between the vector of predictions $\hat{\mathbf{y}} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n))$ and the vector of observed target values $\mathbf{y} = (y_1, y_2, \ldots, y_n)$. A simple example illustrating the linear regression problem is shown in Figure 1.

To minimize the sum of squared errors, we shall first re-write $E(\mathbf{w})$ as

$$\begin{aligned} E(\mathbf{w}) &= \sum_{i=1}^{n} \left(f(\mathbf{x}_i) - y_i\right)^2 \\ &= \sum_{i=1}^{n} \left(\sum_{j=0}^{k} w_j x_{ij} - y_i\right)^2, \end{aligned}$$

where, again, we expanded each data point $\mathbf{x}_i$ by $x_{i0} = 1$ to simplify the expression.

We can now calculate the gradient $\nabla E(\mathbf{w})$ and the Hessian matrix $H_{E(\mathbf{w})}$. Finding weights for which $\nabla E(\mathbf{w}) = \mathbf{0}$ will result in an extremum while a positive semi-definite Hessian will ensure that the extremum is a minimum. Now, we set the partial derivatives to 0 and solve the equations for each weight w_j

$$\begin{aligned} \frac{\partial E}{\partial w_0} &= 2 \sum_{i=1}^{n} \left(\sum_{j=0}^{k} w_j x_{ij} - y_i\right) x_{i0} = 0 \\ \frac{\partial E}{\partial w_1} &= 2 \sum_{i=1}^{n} \left(\sum_{j=0}^{k} w_j x_{ij} - y_i\right) x_{i1} = 0 \end{aligned}$$

$$\vdots$$

$$\frac{\partial E}{\partial w_k} = 2 \sum_{i=1}^{n} \left(\sum_{j=0}^{k} w_j x_{ij} - y_i \right) x_{ik} = 0$$

This results in a system of $k+1$ linear equations with $k+1$ unknowns that can be routinely solved (e.g. by using Gaussian elimination).

While this formulation is useful, it does not allow us to obtain a closed-form solution for $\mathbf{w}$ or discuss the existence or multiplicity of solutions. To address the first point we will exercise some matrix calculus, while the remaining points will be discussed later. We will first write the sum of square errors using the matrix notation as

$$\begin{aligned} E(\mathbf{w}) &= (\mathbf{X}\mathbf{w} - \mathbf{y})^T (\mathbf{X}\mathbf{w} - \mathbf{y}) \\ &= \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 , \end{aligned}$$

where $\|\mathbf{v}\| = \sqrt{\mathbf{v}^T\mathbf{v}} = \sqrt{v_1^2 + v_2^2 + \ldots v_n^2}$ is the length of vector $\mathbf{v}$; it is also called the L_2 norm. We can now formalize the *ordinary least-squares (OLS) linear regression problem* as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\| .$$

We proceed by finding $\nabla E(\mathbf{w})$ and $H_{E(\mathbf{w})}$. The gradient function $\nabla E(\mathbf{w})$ is a derivative of a scalar with respect to a vector. However, the intermediate steps of calculating the gradient require derivatives of vectors with respect to vectors (some of the rules of such derivatives are shown in Table 1). Application of the rules from Table 1 results in

$$\nabla E(\mathbf{w}) = 2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^T\mathbf{y}$$

and, therefore, from $\nabla E(\mathbf{w}) = \mathbf{0}$ we find that

$$\mathbf{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{1}$$

The next step is to find the second derivative in order to ensure that we have not found a maximum. This results in

$$H_{E(\mathbf{w})} = 2\mathbf{X}^T\mathbf{X},$$

which is a positive semi-definite matrix (why? Consider that for any vector $\mathbf{x} \neq \mathbf{0}$, $\mathbf{x}^T\mathbf{A}^T\mathbf{A}\mathbf{x} = (\mathbf{A}\mathbf{x})^T\mathbf{A}\mathbf{x} = \|\mathbf{A}\mathbf{x}\|^2 \geq 0$, with equality only if the columns of $\mathbf{A}$ are linearly dependent). Thus, we indeed have found a minimum. This is the global minimum because positive semi-definite Hessian implies convexity

$\mathbf{y}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$
$\mathbf{Ax}$	$\mathbf{A}^T$
$\mathbf{x}^T\mathbf{A}$	$\mathbf{A}$
$\mathbf{x}^T\mathbf{x}$	$2\mathbf{x}$
$\mathbf{x}^T\mathbf{Ax}$	$\mathbf{Ax} + \mathbf{A}^T\mathbf{x}$

Table 1: Useful derivative formulas of vectors with respect to vectors. The derivative of vector $\mathbf{y}$ (say m-dimensional) with respect to vector $\mathbf{x}$ (say n-dimensional) is an $n \times m$ matrix $\mathbf{M}$ with components $M_{ij} = {}^{\partial y_j}/_{\partial x_i}$, $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, m\}$. A derivative of scalar with respect to a vector, e.g. the gradient, is a special case of this situation that results in an $n \times 1$ column vector.

of $E(\mathbf{w})$. Furthermore, if the columns of $\mathbf{X}$ are linearly independent, Hessian is positive definite, which implies that the global minimum is unique. We can now express the predicted target values as

$$\begin{aligned}\hat{\mathbf{y}} &= \mathbf{X}\mathbf{w}^* \\ &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y},\end{aligned}$$

which has $O(n^3)$ time complexity, assuming that n and k are roughly equal. Matrix $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is called the *projection matrix*, but we will see later what it projects ($\mathbf{y}$) and where (to the column space of $\mathbf{X}$).

Example: Consider again data set $\mathcal{D} = \{(1, 1.2), (2, 2.3), (3, 2.3), (4, 3.3)\}$ from Figure 1. We want to find the optimal coefficients of the least-squares fit for $f(x) = w_0 + w_1 x$ and then calculate the sum of squared errors on $\mathcal{D}$ after the fit.

The OLS fitting can now be performed using

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \ \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} 1.2 \\ 2.3 \\ 2.3 \\ 3.3 \end{bmatrix},$$

where a column of ones was added to $\mathbf{X}$ to allow for a non-zero intercept ($y = w_0$ when $x = 0$). Substituting $\mathbf{X}$ and $\mathbf{y}$ into Eq. (1) results in $\mathbf{w}^* = (0.7, 0.63)$ and the sum of square errors is $E(\mathbf{w}^*) = 0.223$. □

As seen in the example above, it is a standard practice to add a column of ones to the data matrix $\mathbf{X}$ in order to ensure that the fitted line, or generally a hyperplane, does not have to pass through the origin of the coordinate system. This effect, however, can be achieved in other ways. Consider the first component of the gradient vector

$$\frac{\partial E}{\partial w_0} = 2\sum_{i=1}^{n}\left(\sum_{j=0}^{k} w_j x_{ij} - y_i\right) x_{i0} = 0$$

from which we obtain that

$$\sum_{i=1}^{n} w_0 = \sum_{i=1}^{n} y_i - \sum_{j=1}^{k} w_j \sum_{i=1}^{n} x_{ij}.$$

When all features (columns of $\mathbf{X}$) are normalized to have zero mean, i.e. when $\sum_{i=1}^{n} x_{ij} = 0$ for any column j, it follows that

$$w_0 = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

We see now that if the target variable is normalized to the zero mean as well, it follows that $w_0 = 0$ and that the column of ones is not needed.

1.1 Weighted error function

In some applications it is useful to consider minimizing the weighted error function

$$E(\mathbf{w}) = \sum_{i=1}^{n} c_i \left(\sum_{j=0}^{k} w_j x_{ij} - y_i\right)^2,$$

where $c_i > 0$ is a cost for data point i. Expressing this in a matrix form, the goal is to minimize $(\mathbf{X}\mathbf{w} - \mathbf{y})^T \mathbf{C} (\mathbf{X}\mathbf{w} - \mathbf{y})$, where $\mathbf{C} = \mathrm{diag}(c_1, c_2, \ldots, c_n)$. Using a similar approach as above, it can be shown that the weighted least-squares solution $\mathbf{w}_C$ can be expressed as

$$\mathbf{w}_C = \left(\mathbf{X}^T\mathbf{C}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{C}\mathbf{y}.$$

In addition, it can be derived that

$$\mathbf{w}_C = \mathbf{w}_{OLS} + \left(\mathbf{X}^T\mathbf{C}\mathbf{X}\right)^{-1}\mathbf{X}^T(\mathbf{I} - \mathbf{C})(\mathbf{X}\mathbf{w}_{OLS} - \mathbf{y}),$$

where $\mathbf{w}_{OLS}$ is provided by Eq. (1). We can see that the solutions are identical when $\mathbf{C} = \mathbf{I}$, but also when $\mathbf{X}\mathbf{w}_{OLS} = \mathbf{y}$.

2 The Maximum Likelihood Approach

We now consider a statistical formulation of linear regression. We shall first lay out the assumptions behind this process and subsequently formulate the

problem through maximization of the conditional likelihood function. We will then analyze the solution and its basic statistical properties.

Let us assume that the observed data set $\mathcal{D}$ is a product of a data generating process in which n data points were drawn independently and according to the same distribution $p(\mathbf{x})$. Assume also that the target variable Y has an underlying linear relationship with features $(X_1, X_2, \ldots, X_k)$, modified by some error term ε that follows a zero-mean Gaussian distribution, i.e. $\varepsilon : N(0, \sigma^2)$. That is, for a given input $\mathbf{x}$, the target y is a realization of a random variable Y defined as

$$Y = \sum_{j=0}^{k} \omega_j X_j + \varepsilon,$$

where $\boldsymbol{\omega} = (\omega_0, \omega_1, \ldots, \omega_k)$ is a set of unknown coefficients we seek to recover through estimation. Generally, the assumption of normality for the error term is reasonable (recall the central limit theorem!), although the independence between ε and $\mathbf{x}$ may not hold in practice. Using a few simple properties of expectations, we can see that Y also follows a Gaussian distribution, i.e. its conditional density is $p(y|\mathbf{x}, \boldsymbol{\omega}) = N(\boldsymbol{\omega}^T\mathbf{x}, \sigma^2)$.

In linear regression, we seek to approximate the target as $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, where weights $\mathbf{w}$ are to be determined. We first write the conditional likelihood function for a single pair $(\mathbf{x}, y)$ as

$$p(y|\mathbf{x}, \mathbf{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y - \sum_{j=0}^{k} w_j x_j\right)^2}{2\sigma^2}}$$

and observe that the only change from the conditional density function of Y is that coefficients $\mathbf{w}$ are used instead of $\boldsymbol{\omega}$. Incorporating the entire data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, we can now write the conditional likelihood function as $p(\mathbf{y}|\mathbf{X}, \mathbf{w})$ and find weights as

$$\mathbf{w}^*_{ML} = \arg\max_{\mathbf{w}} \left\{p(\mathbf{y}|\mathbf{X}, \mathbf{w})\right\}.$$

Since examples $\mathbf{x}$ are independent and identically distributed (i.i.d.), we have

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \mathbf{w}) &= \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_i - \sum_{j=0}^{k} w_j x_{ij}\right)^2}{2\sigma^2}}. \end{aligned}$$

For the reasons of mathematical convenience, we will look at the logarithm (monotonic function) of the likelihood function and express the log-likelihood as

$$\ln(p(\mathbf{y}|\mathbf{X},\mathbf{w})) = -\sum_{i=1}^{n}\log\left(\sqrt{2\pi\sigma^2}\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \sum_{j=0}^{k} w_j x_{ij}\right)^2 .$$

Given that the first term on the right-hand hand side is independent of $\mathbf{w}$, maximizing the likelihood function corresponds exactly to minimizing the sum of squared errors $E(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^T (\mathbf{X}\mathbf{w} - \mathbf{y})$. This leads to the solution from Eq. (1).

While we arrived at the same solution, the statistical framework provides new insights into the OLS regression. In particular, the assumptions behind the process are now clearly stated; e.g. the i.i.d. process according to which $\mathcal{D}$ was drawn, underlying linear relationship between features and the target, zero-mean Gaussian noise for the error term and its independence of the features. We also assumed absence of noise in the collection of features.

2.1 Expectation and variance for the solution vector

Under the data generating model presented above, we shall now treat the solution vector (estimator) $\mathbf{w}^*$ as a random variable and investigate its statistical properties. For each of the data points, we have $Y_i = \sum_{j=0}^{k} \omega_j X_{ij} + \varepsilon_i$, and use $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$ to denote a vector of i.i.d. random variables each drawn according to $N(0, \sigma^2)$. Let us now look at the expected value (with respect to training data set D) for the weight vector $\mathbf{w}^*$:

$$\begin{aligned} E_D[\mathbf{w}^*] &= E\left[\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\left(\mathbf{X}\boldsymbol{\omega} + \boldsymbol{\varepsilon}\right)\right] \\ &= \boldsymbol{\omega}, \end{aligned}$$

since $E[\boldsymbol{\varepsilon}] = \mathbf{0}$. An estimator whose expected value is the true value of the parameter is called an *unbiased estimator*. The covariance matrix for the optimal set of parameters can be expressed as

$$\begin{aligned} \Sigma[\mathbf{w}^*] &= E\left[\left(\mathbf{w}^* - \boldsymbol{\omega}\right)\left(\mathbf{w}^* - \boldsymbol{\omega}\right)^T\right] \\ &= E\left[\mathbf{w}^*\mathbf{w}^{*T}\right] - \boldsymbol{\omega}\boldsymbol{\omega}^T \end{aligned}$$

Taking $\mathbf{X}^\dagger = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$, we have

$$\begin{aligned} \Sigma[\mathbf{w}^*] &= E\left[\left(\boldsymbol{\omega} + \mathbf{X}^\dagger\boldsymbol{\varepsilon}\right)\left(\boldsymbol{\omega} + \mathbf{X}^\dagger\boldsymbol{\varepsilon}\right)^T\right] - \boldsymbol{\omega}\boldsymbol{\omega}^T \\ &= \boldsymbol{\omega}\boldsymbol{\omega}^T + E\left[\mathbf{X}^\dagger\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\mathbf{X}^{\dagger T}\right] - \boldsymbol{\omega}\boldsymbol{\omega}^T \\ &= \mathbf{X}^\dagger E\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\right]\mathbf{X}^{\dagger T} \end{aligned}$$

because $\mathbf{X}^\dagger\mathbf{X} = \mathbf{I}$ and $E[\boldsymbol{\varepsilon}] = \mathbf{0}$. Thus, we have

$$\Sigma[\mathbf{w}^*] = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\sigma^2,$$

where we exploited the facts that $E\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\right] = \sigma^2\mathbf{I}$ and that an inverse of a symmetric matrix is also a symmetric matrix. It can be shown that estimator $\mathbf{w}^* = \mathbf{X}^\dagger\mathbf{y}$ is the one with the smallest variance among all unbiased estimators (Gauss-Markov theorem). These results are important not only because the estimated coefficients are expected to match the true coefficients of the underlying linear relationship, but also because the matrix inversion in the covariance matrix suggests stability problems in cases of singular and nearly singular matrices. Numerical stability and sensitivity to perturbations will be discussed later.

3 An Algebraic Perspective

Another powerful tool for analyzing and understanding linear regression comes from linear and applied linear algebra. In this section we take a detour to address fundamentals of linear algebra and then apply these concepts to deepen our understanding of regression. In linear algebra, we are frequently interested in solving the following set of equations, given below in a matrix form

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{2}$$

Here, $\mathbf{A}$ is an $m \times n$ matrix, $\mathbf{b}$ is an $m \times 1$ vector, and $\mathbf{x}$ is an $n \times 1$ vector that is to be found. All elements of $\mathbf{A}$, $\mathbf{x}$, and $\mathbf{b}$ are considered to be real numbers. We shall start with a simple scenario and assume $\mathbf{A}$ is a square, 2×2 matrix. This set of equations can be expressed as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2 \end{aligned}$$

For example, we may be interested in solving

$$\begin{aligned} x_1 + 2x_2 &= 3 \\ x_1 + 3x_2 &= 5 \end{aligned}$$

This is a convenient formulation when we want to solve the system, e.g. by Gaussian elimination. However, it is not a suitable formulation to understand the question of the existence of solutions. In order for us to do this, we briefly review the basic concepts in linear algebra.

3.1 The four fundamental subspaces

The objective of this section it to briefly review the *four fundamental subspaces* in linear algebra (column space, row space, nullspace, left nullspace) and their mutual relationship. We shall start with our example from above and write the system of linear equations as

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} x_1 + \begin{bmatrix} 2 \\ 3 \end{bmatrix} x_2 = \begin{bmatrix} 3 \\ 5 \end{bmatrix}.$$

We can see now that by solving $\mathbf{Ax} = \mathbf{b}$ we are looking for the right amounts of vectors $(1, 1)$ and $(2, 3)$ so that their linear combination produces $(3, 5)$; these amounts are $x_1 = -1$ and $x_2 = 2$. Let us define $\mathbf{a}_1 = (1, 1)$ and $\mathbf{a}_2 = (2, 3)$ to be the column vectors of $\mathbf{A}$; i.e. $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2]$. Thus, $\mathbf{Ax} = \mathbf{b}$ will be solvable whenever $\mathbf{b}$ can be expressed as a linear combination of the column vectors $\mathbf{a}_1$ and $\mathbf{a}_2$.

All linear combinations of the columns of matrix $\mathbf{A}$ constitute the *column space* of $\mathbf{A}$, $C(\mathbf{A})$, with vectors $\mathbf{a}_1 \dots \mathbf{a}_n$ being a basis of this space. Both $\mathbf{b}$ and $C(\mathbf{A})$ lie in the m-dimensional space $\mathbb{R}^m$. Therefore, what $\mathbf{Ax} = \mathbf{b}$ is saying is that $\mathbf{b}$ must lie in the column space of $\mathbf{A}$ for the equation to have solutions. In the example above, if columns of $\mathbf{A}$ are linearly independent (as a reminder, two vectors are independent when their linear combination cannot be zero, unless both x_1 and x_2 are zero), the solution is unique, i.e. there exists only one linear combination of the column vectors that will give $\mathbf{b}$. Otherwise, because $\mathbf{A}$ is a square matrix, the system has no solutions. An example of such a situation is

$$\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \end{bmatrix},$$

where $\mathbf{a}_1 = (1, 1)$ and $\mathbf{a}_2 = (2, 2)$. Here, $\mathbf{a}_1$ and $\mathbf{a}_2$ are (linearly) dependent because $2\mathbf{a}_1 - \mathbf{a}_2 = 0$. There is a deep connection between the spaces generated by a set of vectors and the properties of the matrix A. For now, using the example above, it suffices to say that if $\mathbf{a}_1$ and $\mathbf{a}_2$ are independent the matrix $\mathbf{A}$ is non-singular (singularity can be discussed only for square matrices), that is of full rank.

In an equivalent manner to the column space, all linear combinations of the rows of $\mathbf{A}$ constitute the *row space*, denoted by $C(\mathbf{A}^T)$, where both $\mathbf{x}$ and $C(\mathbf{A}^T)$ are in $\mathbb{R}^n$. All solutions to $\mathbf{Ax} = \mathbf{0}$ constitute the *nullspace* of the matrix, $N(\mathbf{A})$, while all solutions of $\mathbf{A}^T\mathbf{y} = \mathbf{0}$ constitute the so-called *left nullspace* of $\mathbf{A}$, $N(\mathbf{A}^T)$. Clearly, $C(\mathbf{A})$ and $N(\mathbf{A}^T)$ are embedded in $\mathbb{R}^m$, whereas $C(\mathbf{A}^T)$ and $N(\mathbf{A})$ are in $\mathbb{R}^n$. However, the pairs of subspaces are orthogonal (vectors $\mathbf{u}$ and $\mathbf{v}$ are orthogonal if $\mathbf{u}^T\mathbf{v} = 0$); that is, any vector in $C(\mathbf{A})$ is orthogonal to all vectors from $N(\mathbf{A}^T)$ and any vector in $C(\mathbf{A}^T)$ is orthogonal to all vectors from $N(\mathbf{A})$. This is easy to see: if $\mathbf{x} \in N(\mathbf{A})$, then by definition $\mathbf{Ax} = \mathbf{0}$, and thus each row of $\mathbf{A}$ is orthogonal to $\mathbf{x}$. If each row is orthogonal to $\mathbf{x}$, then so are all linear combinations of rows.

Orthogonality is a key property of the four subspaces, as it provides useful decomposition of vectors $\mathbf{x}$ and $\mathbf{b}$ from Eq. (2) with respect to $\mathbf{A}$ (we will exploit

this in the next Section). For example, any $\mathbf{x} \in \mathbb{R}^n$ can be decomposed as

$$\mathbf{x} = \mathbf{x}_r + \mathbf{x}_n,$$

where $\mathbf{x}_r \in C(\mathbf{A}^T)$ and $\mathbf{x}_n \in N(\mathbf{A})$, such that $\|\mathbf{x}\|^2 = \|\mathbf{x}_r\|^2 + \|\mathbf{x}_n\|^2$. Similarly, every $\mathbf{b} \in \mathbb{R}^m$ can be decomposed as

$$\mathbf{b} = \mathbf{b}_c + \mathbf{b}_l,$$

where $\mathbf{b}_c \in C(\mathbf{A})$, $\mathbf{b}_l \in N(\mathbf{A}^T)$, and $\|\mathbf{b}\|^2 = \|\mathbf{b}_c\|^2 + \|\mathbf{b}_l\|^2$.

We mentioned above that the properties of fundamental spaces are tightly connected with the properties of matrix $\mathbf{A}$. To conclude this section, let us briefly discuss the *rank* of a matrix and its relationship with the dimensions of the fundamental subspaces. The *basis* of the space is the smallest set of vectors that span the space (this set of vectors is not unique). The size of the basis is also called the dimension of the space. In the example at the beginning of this subsection, we had a two dimensional column space with basis vectors $\mathbf{a}_1 = (1, 1)$ and $\mathbf{a}_2 = (2, 3)$. On the other hand, for $\mathbf{a}_1 = (1, 1)$ and $\mathbf{a}_2 = (2, 2)$ we had a one dimensional column space, i.e. a line, fully determined by any of the basis vectors. Unsurprisingly, the dimension of the space spanned by column vectors equals the *rank* of matrix $\mathbf{A}$. One of the fundamental results in linear algebra is that the rank of $\mathbf{A}$ is identical to the dimension of $C(\mathbf{A})$, which in turn is identical to the dimension of $C(\mathbf{A}^T)$.

3.2 Minimizing $\|\mathbf{Ax} - \mathbf{b}\|$

Let us now look again at the solutions to $\mathbf{Ax} = \mathbf{b}$. In general, there are three different outcomes:

1. there are no solutions to the system
2. there is a unique solution to the system, and
3. there are infinitely many solutions.

These outcomes depend on the relationship between the rank (r) of $\mathbf{A}$ and dimensions m and n. We already know that when $r = m = n$ (square, invertible, full rank matrix $\mathbf{A}$) there is a unique solution to the system, but let us investigate other situations. Generally, when $r = n < m$ (full column rank), the system has either one solution or no solutions, as we will see momentarily. When $r = m < n$ (full row rank), the system has infinitely many solutions. Finally, in cases when $r < m$ and $r < n$, there are either no solutions or there are infinitely many solutions. Because $\mathbf{Ax} = \mathbf{b}$ may not be solvable, we generalize solving $\mathbf{Ax} = \mathbf{b}$ to minimizing $\|\mathbf{Ax} - \mathbf{b}\|$. In such a way, all situations can be considered in a unified framework.

Let us consider the following example

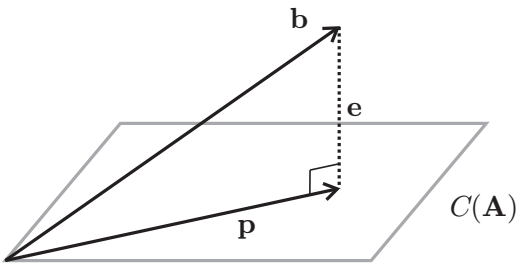

Figure 2: Illustration of the projection of vector $\mathbf{b}$ to the column space of matrix $\mathbf{A}$. Vectors $\mathbf{p}$ ($\mathbf{b}_c$) and $\mathbf{e}$ ($\mathbf{b}_l$) represent the projection point and the error, respectively.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix},$$

which illustrates an instance where we are unlikely to have a solution to $\mathbf{Ax} = \mathbf{b}$, unless there is some constraint on b_1, b_2, and b_3; here, the constraint is $b_3 = 2b_2 - b_1$. In this situation, $C(\mathbf{A})$ is a 2D plane in $\mathbb{R}^3$ spanned by the column vectors $\mathbf{a}_1 = (1, 1, 1)$ and $\mathbf{a}_2 = (2, 3, 4)$. If the constraint on the elements of $\mathbf{b}$ is not satisfied, our goal is to try to find a point in $C(\mathbf{A})$ that is closest to $\mathbf{b}$**.** This happens to be the point where $\mathbf{b}$ is projected to $C(\mathbf{A})$, as shown in Figure 2. We will refer to the projection of $\mathbf{b}$ to $C(\mathbf{A})$ as $\mathbf{p}$. Now, using the standard algebraic notation, we have the following equations

$$\begin{aligned} \mathbf{b} &= \mathbf{p} + \mathbf{e} \\ \mathbf{p} &= \mathbf{Ax} \end{aligned}$$

Since $\mathbf{p}$ and $\mathbf{e}$ are orthogonal, we know that $\mathbf{p}^T\mathbf{e} = 0$. Let us now solve for $\mathbf{x}$

$$\begin{aligned} (\mathbf{Ax})^T(\mathbf{b} - \mathbf{Ax}) &= 0 \\ \mathbf{x}^T\mathbf{A}^T\mathbf{b} - \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} &= 0 \\ \mathbf{x}^T\left(\mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{Ax}\right) &= 0 \end{aligned}$$

and thus

$$\mathbf{x}^* = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}.$$

This is exactly the same solution as one that minimized the sum of squared errors and maximized the likelihood. Matrix

$$\mathbf{A}^\dagger = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$$

is called the Moore-Penrose pseudo-inverse or simply a pseudo-inverse. This is an important matrix because it always exists and is unique, even in situations

when the inverse of $\mathbf{A}^T\mathbf{A}$ does not exist. This happens when $\mathbf{A}$ has dependent columns (technically, $\mathbf{A}$ and $\mathbf{A}^T\mathbf{A}$ will have the same nullspace that contains more than just the origin of the coordinate system; thus the rank of $\mathbf{A}^T\mathbf{A}$ is less than n). Let us for a moment look at the projection vector $\mathbf{p}$. We have

$$\begin{aligned}\mathbf{p} &= \mathbf{A}\mathbf{x} \\ &= \mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b},\end{aligned}$$

where $\mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$ is the matrix that projects $\mathbf{b}$ to the column space of $\mathbf{A}$.

While we arrived at the same result as in previous sections, the tools of linear algebra allow us to discuss OLS regression at a deeper level. Let us examine for a moment the existence and multiplicity of solutions to

$$\arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|. \tag{3}$$

Clearly, the solution to this problem always exists. However, we shall now see that the solution to this problem is generally not unique and that it depends on the rank of $\mathbf{A}$. Consider $\mathbf{x}$ to be one solution to Eq. (3). Recall that $\mathbf{x} = \mathbf{x}_r + \mathbf{x}_n$ and that it is multiplied by $\mathbf{A}$; thus any vector $\mathbf{x} = \mathbf{x}_r + \alpha\mathbf{x}_n$, where $\alpha \in \mathbb{R}$, is also a solution. Observe that $\mathbf{x}_r$ is common to all such solutions; if you cannot see it, assume there exists another vector from the row space and show that it is not possible. If the columns of $\mathbf{A}$ are independent, the solution is unique because the nullspace contains only the origin. Otherwise, there are infinitely many solutions. In such cases, what exactly is the solution found by projecting $\mathbf{b}$ to $C(\mathbf{A})$? Let us look at it:

$$\begin{aligned}\mathbf{x}^* &= \mathbf{A}^\dagger\mathbf{b} \\ &= \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T(\mathbf{p}+\mathbf{e}) \\ &= \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{p} \\ &= \mathbf{x}_r,\end{aligned}$$

as $\mathbf{p} = \mathbf{A}\mathbf{x}_r$. Given that $\mathbf{x}_r$ is unique, the solution found by the least squares optimization is the one that simultaneously minimizes $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|$ and $\|\mathbf{x}\|$ (observe that $\|\mathbf{x}\|$ is minimized because the solution ignores any component from the nullspace). Thus, the OLS regression problem is sometimes referred to as the minimum-norm least-squares problem.

Let us now consider situations where $\mathbf{A}\mathbf{x} = \mathbf{b}$ has infinitely many solutions, i.e. when $\mathbf{b} \in C(\mathbf{A})$. This usually arises when $r \leq m < n$. Here, because $\mathbf{b}$ is already in the column space of $\mathbf{A}$, the only question is what particular solution $\mathbf{x}$ will be found by the minimization procedure. As we have seen above, the outcome of the minimization process is the solution with the minimum L_2 norm $\|\mathbf{x}\|$.

To summarize, the goal of the OLS regression problem is to solve $\mathbf{X}\mathbf{w} = \mathbf{y}$ if it is solvable. When $k < n$ this is not a realistic scenario in practice. Thus,

we relaxed the requirement and tried to find the point in the column space $C(\mathbf{X})$ that is closest to $\mathbf{y}$. This turned out to be equivalent to minimizing the sum of square errors (or Euclidean distance) between n-dimensional vectors $\mathbf{Xw}$ and $\mathbf{y}$. It also turned out to be equivalent to the maximum likelihood solution presented in Section 2. When $n < k$, a usual situation in practice is that there are infinitely many solutions. In these situations, our optimization algorithm will find the one with the minimum L_2 norm.

4 Linear Regression for Non-Linear Problems

At first, it might seem that the applicability of linear regression to real-life problems is greatly limited. After all, it is not clear whether it is realistic (most of the time) to assume that the target variable is a linear combination of features. Fortunately, the applicability of linear regression is broader than originally thought. The main idea is to apply a non-linear transformation to the data matrix $\mathbf{X}$ prior to the fitting step, which then enables a non-linear fit. In this section we examine two applications of linear regression: polynomial curve fitting and radial basis function (RBF) networks.

4.1 Polynomial curve fitting

We start with one-dimensional data. In OLS regression, we would look for the fit in the following form

$$f(x) = w_0 + w_1 x,$$

where x is the data point and $\mathbf{w} = (w_0, w_1)$ is the weight vector. To achieve a polynomial fit of degree p, we will modify the previous expression into

$$f(x) = \sum_{j=0}^{p} w_j x^j,$$

where p is the degree of the polynomial. We will rewrite this expression using a set of basis functions as

$$\begin{aligned} f(x) &= \sum_{j=0}^{p} w_j \phi_j(x) \\ &= \mathbf{w}^T \boldsymbol{\phi}, \end{aligned}$$

where $\phi_j(x) = x^j$ and $\boldsymbol{\phi} = (\phi_0(x), \phi_1(x), \ldots, \phi_p(x))$. Applying this transformation to every data point in $\mathbf{X}$ results in a new data matrix $\mathbf{\Phi}$, as shown in Figure 3.

Following the discussion from Section 1, the optimal set of weights is now calculated as

$$\mathbf{w}^* = \left(\mathbf{\Phi}^T \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^T \mathbf{y}.$$

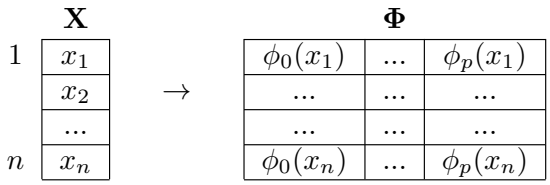

Figure 3: Transformation of an $n \times 1$ data matrix $\mathbf{X}$ into an $n \times (p+1)$ matrix $\mathbf{\Phi}$ using a set of basis functions ϕ_j, $j = 0, 1, \ldots, p$.

Example: In Figure 1 we presented an example of a data set with four data points. What we did not mention was that, given a set $\{x_1, x_2, x_3, x_4\}$, the targets were generated by using function $1 + \frac{x}{2}$ and then adding a measurement error $\mathbf{e} = (-0.3, 0.3, -0.2, 0.3)$. It turned out that the optimal coefficients $\mathbf{w}^* = (0.7, 0.63)$ were close to the true coefficients $\boldsymbol{\omega} = (1, 0.5)$, even though the error terms were relatively significant. We will now attempt to estimate the coefficients of a polynomial fit with degrees $p = 2$ and $p = 3$. We will also calculate the sum of squared errors on $\mathcal{D}$ after the fit as well as on a large discrete set of values $x \in \{0, 0.1, 0.2, \ldots, 10\}$ where the target values will be generated using the true function $1 + \frac{x}{2}$.

Using a polynomial fit with degrees $p = 2$ and $p = 3$ results in $\mathbf{w}_2^* = (0.575, 0.755, -0.025)$ and $\mathbf{w}_3^* = (-3.1, 6.6, -2.65, 0.35)$, respectively. The sums of squared errors on $\mathcal{D}$ equal $E(\mathbf{w}_2^*) = 0.221$ and $E(\mathbf{w}_3^*) \approx 0$. Thus, the best fit is achieved with the cubic polynomial. However, the sums of squared errors on the outside data set reveal a poor generalization ability of the cubic model because we obtain $E(\mathbf{w}^*) = 26.9$, $E(\mathbf{w}_2^*) = 3.9$, and $E(\mathbf{w}_3^*) = 22018.5$. This effect is called *overfitting*. Broadly speaking, overfitting is indicated by a significant difference in fit between the data set on which the model was trained and the outside data set on which the model is expected to be applied (Figure 4). In this case, the overfitting occurred because the complexity of the model was increased considerably, whereas the size of the data set remained small.

One signature of overfitting is an increase in the magnitude of the coefficients. For example, while the absolute values of all coefficients in $\mathbf{w}^*$ and $\mathbf{w}_2^*$ were less than one, the values of the coefficients in $\mathbf{w}_3^*$ became significantly larger with alternating signs (suggesting overcompensation). A particular form of learning, the one using regularized error functions, is developed to prevent this effect. □

Polynomial curve fitting is only one way of non-linear fitting because the choice of basis functions need not be limited to powers of x. Among others, non-linear basis functions that are commonly used are the sigmoid function

$$\phi_j(x) = \frac{1}{1 + e^{-\frac{x - \mu_j}{s_j}}}$$

or a Gaussian-style exponential function

$$\phi_j(x) = e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}},$$

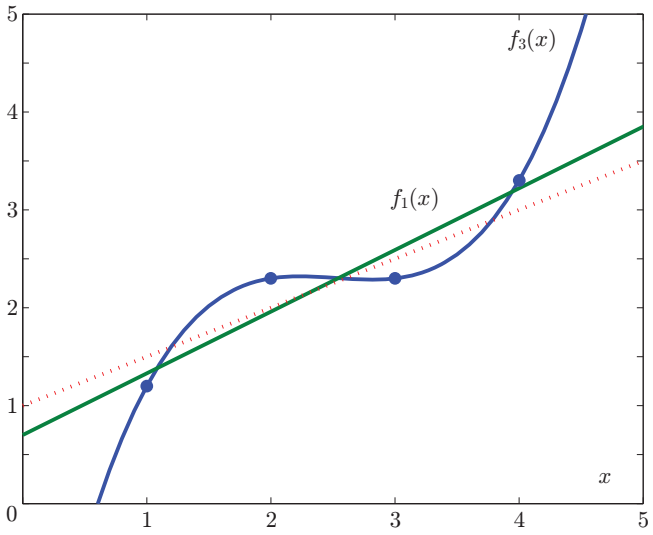

Figure 4: Example of a linear vs. polynomial fit on a data set shown in Figure 1. The linear fit, $f_1(x)$, is shown as a solid green line, whereas the cubic polynomial fit, $f_3(x)$, is shown as a solid blue line. The dotted red line indicates the target linear concept.

where μ_j, s_j, and σ_j are constants to be determined. However, this approach works only for a one-dimensional input $\mathbf{X}$. For higher dimensions, this approach needs to be generalized using *radial basis functions*.

4.2 Radial basis function networks

The idea of radial basis function (RBF) networks is a natural generalization of the polynomial curve fitting and approaches from the previous Section. Given data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, we start by picking p points to serve as the "centers" in the input space $\mathcal{X}$. We denote those centers as $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_p$. Usually, these can be selected from $\mathcal{D}$ or computed using some clustering technique (e.g. the EM algorithm, K-means).

When the clusters are determined using a Gaussian mixture model, the basis functions can be selected as

$$\phi_j(\mathbf{x}) \quad = \quad e^{-\frac{1}{2}(\mathbf{x}-\mathbf{c}_j)^T \Sigma_j^{-1} (\mathbf{x}-\mathbf{c}_j)},$$

where the cluster centers and the covariance matrix are found during clustering. When K-means or other clustering is used, we can use

$$\phi_j(\mathbf{x}) = e^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{2\sigma_j^2}},$$

where σ_j's can be separately optimized; e.g. using a validation set. In the

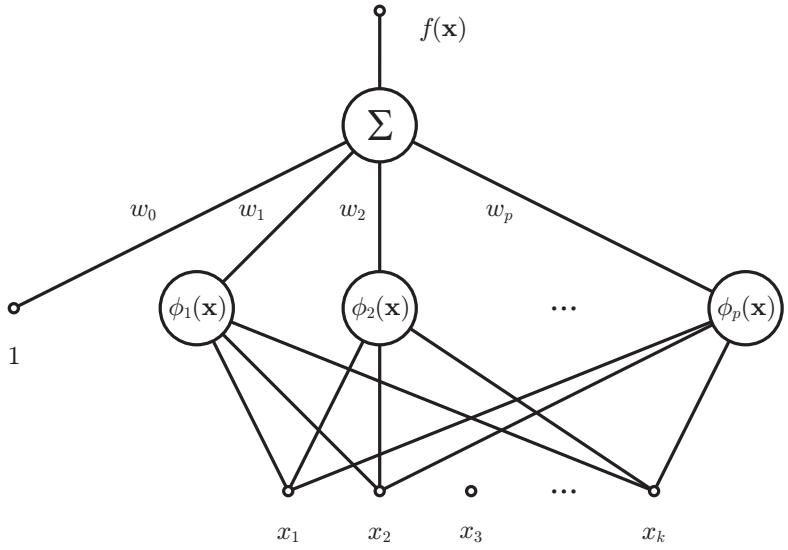

Figure 5: Radial basis function network.

context of multidimensional transformations from $\mathbf{X}$ to $\mathbf{\Phi}$, the basis functions can also be referred to as *kernel functions*, i.e. $\phi_j(\mathbf{x}) = k_j(\mathbf{x}, \mathbf{c}_j)$. Matrix

$$\mathbf{\Phi} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_p(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & & \\ \vdots & & \ddots & \\ \phi_0(\mathbf{x}_n) & & & \phi_p(\mathbf{x}_n) \end{bmatrix}$$

is now used as a new data matrix. For a given input $\mathbf{x}$, the prediction of the target y will be calculated as

$$\begin{aligned} f(\mathbf{x}) &= w_0 + \sum_{j=1}^{p} w_j \phi_j(\mathbf{x}) \\ &= \sum_{j=0}^{p} w_j \phi_j(\mathbf{x}) \end{aligned}$$

where $\phi_0(\mathbf{x}) = 1$ and $\mathbf{w}$ is to be found. It can be proved that with a sufficiently large number of radial basis functions we can accurately approximate any function. As seen in Figure 5, we can think of RBFs as neural networks.

5 Generalized Linear Models

In previous sections, we saw that the statistical framework provided valuable insights into linear regression, especially with respect to explicitly stating most of the assumptions in the system (we will see the full picture only when Bayesian

formulation is used). These assumptions were necessary to rigorously estimate parameters of the model, which could then be subsequently used for prediction on previously unseen data points. In this section, we introduce generalized linear models (GLMs) which extend ordinary least-squares regression beyond Gaussian probability distributions and linear dependencies between the features and the target.

We shall first revisit the main points of the ordinary least-squares regression. There, we assumed that a set of i.i.d. data points with their targets $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ were drawn according to some distribution $p(\mathbf{x}, y)$. We also assumed that an underlying relationship between the features and the target was linear, i.e.

$$Y = \sum_{j=0}^{k} \omega_j X_j + \varepsilon,$$

where $\boldsymbol{\omega}$ was a set of unknown weights and ε was a zero-mean normally distributed random variable with variance σ^2. In order to simplify generalization, we will slightly reformulate this model. In particular, it will be useful to separate the underlying linear relationship between the features and the target from the fact that Y was normally distributed. That is, we will write that

1. $E[y|\mathbf{x}] = \boldsymbol{\omega}^T\mathbf{x}$
2. $p(y|\mathbf{x}) = N(\mu, \sigma^2)$

with $\mu = \boldsymbol{\omega}^T\mathbf{x}$ connecting the two expressions. This way of formulating linear regression will allow us (i) to generalize the framework to non-linear relationships between the features and the target as well as (ii) to use the error distributions other than Gaussian.

5.1 Loglinear link and Poisson distribution

Let us start with an example. Assume that data points correspond to cities in the world (described by some numerical features) and that the target variable is the number of sunny days observed in a particular year. To establish the GLM model, we will assume (1) a loglinear link between the expectation of the target and linear combination of features, and (2) the Poisson distribution for the target variable. We summarize these assumptions as follows

1. $\log(E[y|\mathbf{x}]) = \boldsymbol{\omega}^T\mathbf{x}$
2. $p(y|\mathbf{x}) = \text{Poisson}(\lambda)$

where $\lambda > 0$ is the parameter (mean and variance) of the Poisson distribution. Exploiting the fact that $E[y|\mathbf{x}] = \lambda$, we connect the two formulas using $\lambda = e^{\boldsymbol{\omega}^T\mathbf{x}}$. In fact, because $\lambda \in \mathbb{R}^+$ and $\boldsymbol{\omega}^T\mathbf{x} \in \mathbb{R}$, it is not appropriate to use a linear link between $E[y|\mathbf{x}]$ and $\boldsymbol{\omega}^T\mathbf{x}$ (i.e. $E[y|\mathbf{x}] = \boldsymbol{\omega}^T\mathbf{x}$). The link function adjusts

the range of the linear combination of features (so-called systematic component) to the domain of the parameters (here, mean) of the probability distribution.

We provide a compact summary of the above assumptions via a probability distribution for the target; i.e. $p(y|\mathbf{x}) = \text{Poisson}(e^{\boldsymbol{\omega}^T\mathbf{x}})$. We express this as

$$p(y|\mathbf{x}) = \frac{e^{\boldsymbol{\omega}^T\mathbf{x}y} \cdot e^{-e^{\boldsymbol{\omega}^T\mathbf{x}}}}{y!}$$

for any $y \in \mathbb{N}$. We will now use the maximum likelihood estimation to find the parameters of the regression model. As in previous sections, the likelihood function has the form of the probability distribution, where the data set is observed and the parameters are unknown. Hence, the log-likelihood function has the form

$$ll(\mathbf{w}) = \sum_{i=1}^{n} \mathbf{w}^T\mathbf{x}_i y_i - \sum_{i=1}^{n} e^{\mathbf{w}^T\mathbf{x}_i} - \sum_{i=1}^{n} y_i!$$

It is easy to show that $\nabla ll(\mathbf{w}) = \mathbf{0}$ does not have a closed-form solution. Therefore, we will use the Newton-Raphson method in which we must first analytically find the gradient vector $\nabla ll(\mathbf{w})$ and the Hessian matrix $H_{ll(\mathbf{w})}$. We start by deriving the j-th element of the gradient

$$\begin{aligned}
\frac{\partial ll(\mathbf{w})}{\partial w_j} &= \sum_{i=1}^{n} x_{ij} y_i - \sum_{i=1}^{n} e^{\mathbf{w}^T\mathbf{x}_i} x_{ij} \\
&= \sum_{i=1}^{n} x_{ij} \cdot \left(y_i - e^{\mathbf{w}^T\mathbf{x}_i} \right) \\
&= \mathbf{f}_j^T \cdot (\mathbf{y} - \mathbf{c}),
\end{aligned}$$

where $\mathbf{f}_j^T$ is the j-th column of $\mathbf{X}$ and $\mathbf{c}$ is a vector with elements $c_i = e^{\mathbf{w}^T\mathbf{x}_i}$. The gradient of the likelihood function can now be expressed as

$$\nabla ll(\mathbf{w}) = \mathbf{X}^T \cdot (\mathbf{y} - \mathbf{c}). \tag{4}$$

Note that $\mathbf{c}$ stores a set of predictions for each of the data points, and thus, $\mathbf{y} - \mathbf{c}$ is an error vector. The second partial derivative of the likelihood function can be derived as

$$\begin{aligned}
\frac{\partial^2 ll(\mathbf{w})}{\partial w_j \partial w_k} &= -\sum_{i=1}^{n} x_{ij} \cdot e^{\mathbf{w}^T\mathbf{x}_i} \cdot x_{ik} \\
&= -\mathbf{f}_j^T \cdot \mathbf{C} \cdot \mathbf{f}_k,
\end{aligned}$$

where $\mathbf{C}$ is an n-by-n diagonal matrix with $c_{ii} = e^{\mathbf{w}^T\mathbf{x}_i}$. The Hessian matrix can now be calculated as

$$H_{ll(\mathbf{w})} = -\mathbf{X}^T \cdot \mathbf{C} \cdot \mathbf{X}. \tag{5}$$

which is a negative semi-definite matrix. Substituting Eq. (4) and Eq. (5) into the Newton-Raphson formula results in the following weight update rule

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \left(\mathbf{X}^T \cdot \mathbf{C}^{(t)} \cdot \mathbf{X}\right)^{-1} \cdot \mathbf{X}^T \cdot \left(\mathbf{y} - \mathbf{c}^{(t)}\right),$$

where $\mathbf{c}^{(t)}$ and $\mathbf{C}^{(t)}$ are calculated using the weight vector $\mathbf{w}^{(t)}$. The initial set of weights $\mathbf{w}^{(0)}$ can be set randomly.

5.2 Exponential family of distributions

Exponential family is a class of probability distributions with the following form

$$p(x|\boldsymbol{\theta}) = c(\boldsymbol{\theta})h(x)\exp\left(\sum_{i=1}^{m} q_i(\boldsymbol{\theta})t_i(x)\right),$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m)$ is a set of parameters. When $q_i(\boldsymbol{\theta}) = \theta_i$ for $\forall i$, parameters $\theta_1, \theta_2, \ldots, \theta_m$ are called natural parameters. This leads to the following (canonical) form of the probability distribution

$$\begin{aligned} p(x|\boldsymbol{\theta}) &= \exp\left(\sum_{i=1}^{m} \theta_i t_i(x) - a(\boldsymbol{\theta}) + b(x)\right) \\ &= \exp\left(\boldsymbol{\theta}^T \boldsymbol{t}(x) - a(\boldsymbol{\theta}) + b(x)\right), \end{aligned}$$

where $\boldsymbol{t}(x) = (t_1(x), t_2(x), \ldots, t_m(x))$. Many of the often encountered (families of) distributions are members of the exponential family; e.g. exponential, Gaussian, Gamma, Poisson, or the binomial distributions. Therefore, it is useful to generically study the exponential family to better understand commonalities and differences between individual member functions.

Let us look at a couple of examples. The Poisson distribution can be expressed as

$$p(x|\lambda) = \exp\left(x\log\lambda - \lambda - \log x!\right),$$

where $\lambda \in \mathbb{R}^+$ and $\mathcal{X} = \mathbb{N}_0$. Thus, $\theta = \log\lambda$, $t(x) = x$, $a(\theta) = e^\theta$, and $b(x) = -\log x!$. Similarly, the Gaussian distribution with mean μ and variance σ^2 can be written as

$$p(x|\mu,\sigma) = \exp\left(-\frac{x^2}{2\sigma^2} + x\frac{\mu}{\sigma^2} - \frac{\mu^2}{2\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2)\right),$$

where $\mu \in \mathbb{R}$, $\sigma \in \mathbb{R}^+$ and $\mathcal{X} = \mathbb{R}$. Therefore, we see that

$$
\begin{aligned}
\boldsymbol{\theta} &= (\frac{\mu}{\sigma^2}, -\frac{1}{2\sigma^2}) \\
\boldsymbol{t}(x) &= (x, x^2) \\
a(\boldsymbol{\theta}) &= \frac{\theta_1^2}{4\theta_2} + \frac{1}{2}\log(\frac{\theta_2}{\pi}) \\
b(x) &= 0.
\end{aligned}
$$

From $\int_{\mathcal{X}} p(x)dx = 1$, we can easily derive that

$$
a(\boldsymbol{\theta}) = \log \int_{\mathcal{X}} \exp\left(\boldsymbol{\theta}^T \boldsymbol{t}(x) + b(x)\right) dx.
$$

Function $a(\boldsymbol{\theta})$ is called a log partitioning function or simply a log-normalizer. It can be derived that

$$
\begin{aligned}
\frac{\partial a(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= E\left[\boldsymbol{t}(x)\right] \\
\frac{\partial^2 a(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} &= \mathrm{cov}\left[\boldsymbol{t}(x)\right]
\end{aligned}
$$

These properties are very useful for estimating parameters of the distribution. For example, let us consider a data set of observations $\mathcal{D} = \{x_i\}_{i=1}^n$ and look at the log-likelihood function

$$
\begin{aligned}
ll(\boldsymbol{\theta}) &= \log \prod_{i=1}^{n} e^{\boldsymbol{\theta}^T \boldsymbol{t}(x_i) - a(\boldsymbol{\theta}) + b(x_i)} \\
&= \sum_{i=1}^{n} \boldsymbol{\theta}^T \boldsymbol{t}(x_i) - n \cdot a(\boldsymbol{\theta}) + \sum_{i=1}^{n} b(x_i).
\end{aligned}
$$

Maximizing the likelihood involves calculating the gradient function

$$
\nabla ll(\boldsymbol{\theta}) = \sum_{i=1}^{n} \boldsymbol{t}(x_i) - n \nabla a(\boldsymbol{\theta}).
$$

Setting the gradient to zero results in

$$
\nabla a(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{t}(x_i).
$$

By combining the previous expressions we see that the likelihood is maximized when the gradient function of the log-normalizer equals the sample mean of $\boldsymbol{t}(x)$. This result is important because it provides a general expression for estimating the parameters of all distributions in the exponential family. Function $\boldsymbol{t}(x)$ is called a sufficient statistic (a statistic is simply a function of the data), because all information about the parameters $\boldsymbol{\theta}$ can be inferred by calculating $\boldsymbol{t}(x)$.

5.3 Formalizing generalized linear models

We shall now formalize the generalized linear models. The two key components of GLMs can be expressed as

1. $f(E[y|\mathbf{x}]) = \boldsymbol{\omega}^T\mathbf{x}$
2. $p(y|\mathbf{x}) \in$ Exponential Family

Here, $f(\cdot)$ is called the link function between the linear combination of the features and parameters of the distribution. On the one hand, the link function adjusts the range of $\boldsymbol{\omega}^T\mathbf{x}$ to the domain of Y (because of this relationship, link functions are usually not selected independently of the distribution for Y). On the other hand, it also provides a mechanism for a non-linear relationship between the features and the target. While the nature of this non-linear relationship is limited (the features enter the system via a linear combination with the parameters) there is still an important flexibility they provide for modeling. Similarly, the generalization to the exponential family from the Gaussian distribution used in ordinary least-squares regression, allows us to model a much wider range of target functions. The choice of the link function and the probability distribution is data dependent.

Generally, there is no guarantee of a closed-form solution for $\mathbf{w}$. Therefore, GLM formulations usually resort to iterative techniques derived from the Taylor approximation of the log-likelihood. Hence, a single mechanism can be used for a wide range of link functions and probability distributions. Let us write the log-likelihood

$$\begin{aligned} ll(\mathbf{w}) &= \log \prod_{i=1}^{n} e^{\boldsymbol{\theta}^T \boldsymbol{t}(x_i) - a(\boldsymbol{\theta}) + b(x_i)} \\ &= \sum_i \sum_m \theta_m t_m(x_i) - n \cdot a(\boldsymbol{\theta}) + \sum_i b(x_i) \\ &= \sum_i ll_i(\mathbf{w}) \end{aligned}$$

and also find the elements of its gradient

$$\frac{\partial ll_i(\mathbf{w})}{\partial w_j} = \sum_m \frac{\partial \theta_m}{\partial w_j} t_m(x_i) - \frac{\partial a(\boldsymbol{\theta})}{\partial w_j}$$

which can be used to easily calculate the update rules of the optimization. Interestingly, the standard versions of the GLM from the literature do not use the full version of the Newton-Raphson algorithm with the Hessian matrix. Instead, Gauss-Newton and other types of solutions are considered and are generally called iteratively reweighted least-squares (IRLS) algorithms in the statistical literature.

5.4 Logistic regression

At the end, we mention that GLMs extend to classification. One of the most popular uses of GLMs is a combination of a Bernoulli distribution with a logit link function. This framework is frequently encountered and is called logistic regression. We summarize the logistic regression model as follows

1. $\text{logit}(E[y|\mathbf{x}]) = \boldsymbol{\omega}^T\mathbf{x}$
2. $p(y|\mathbf{x}) = \text{Bernoulli}(\alpha)$

where $\text{logit}(x) = \ln \frac{x}{1-x}$, $y \in \{0, 1\}$, and $\alpha \in (0, 1)$ is the parameter (mean) of the Bernoulli distribution. It follows that

$$E[y|\mathbf{x}] = \frac{1}{1 + e^{-\boldsymbol{\omega}^T\mathbf{x}}}$$

and

$$p(y|\mathbf{x}) = \left(\frac{1}{1 + e^{-\omega^T\mathbf{x}}}\right)^y \left(1 - \frac{1}{1 + e^{-\omega^T\mathbf{x}}}\right)^{1-y}.$$

Given a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^k$ and $y_i \in \{0, 1\}$, the parameters of the model $\mathbf{w}$ can be found by maximizing the likelihood function.