Introduction to Machine Learning (67577)

Recitation 04
Linear Regression 1

Second Semester, 2021

Contents

A regression model is a way to represent a functional relation between a set of explanatory variables (features) in $\mathcal{X}$ and a scalar response, also referred to as dependent variable, in $\mathcal{Y}$. So, we will **assume** that there exists some function $f : \mathcal{X} \to \mathcal{Y}$ that captures this relation for each sample $x \in \mathcal{X}$ and its response $y \in \mathcal{Y}$. This function f is unknown to us and we would like to find it. It may be deterministic or it may contain a random component.

Let's assume first that the relation between $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$ is *deterministic*. So we assume that there exists a function $f : \mathcal{X} \to \mathcal{Y}$ such that, each sample we observe, now or in the future, is of the form $(\mathbf{x}, y)$ with $y = f(\mathbf{x})$. In particular for our training set $y_i = f(\mathbf{x}_i)$ for every training sample $i = 1 \ldots m$. Our goal is to *learn* f from a training sample $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, so we can estimate or predict the value $f(\mathbf{x})$ for a new value $\mathbf{x}$. A sample we haven't seen in our training set – a new sample – is sometimes called a *test* sample. Using the training sample S we will create a function that we hope is as similar as possible to the unknown function f. The function we create is called a **prediction rule** and we denote it by $\hat{f}$ or h_S. (The notation h_S emphasizes that our prediction rule depends on the training sample S).

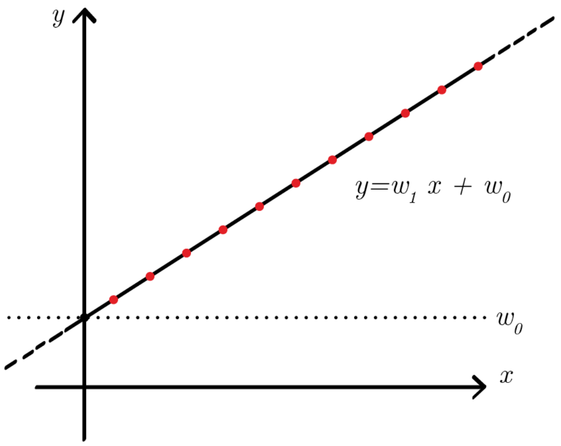

Figure 1: *Illustration of a regression model with $\mathcal{X} = \mathbb{R}$ and $\mathcal{Y} = \mathbb{R}$. Red dots are samples. The solid curve is the learned prediction rule $\hat{f}$.*

For reasons we discuss later (**??**), whenever we try to model a functional relation f, we restrict ourselves to a specific family of functions. Such a family is referred to as a *hypothesis class*. We decide on the hypothesis class before looking at the data, and the prediction rule we find must be in the chosen hypothesis class.

While we can build regression models over various domains $\mathcal{X}$, the simplest domain to consider is the Euclidean space $\mathbb{R}^d$ where each point x is a feature vector with d real numbers. In this chapter and in most of this book, we consider $\mathcal{X} := \mathbb{R}^d$.

0.1 Linear Regression

Let us assume that the relation $\mathcal{X} \to \mathcal{Y}$ is *linear*. This is perhaps the simplest relation we can describe. Formally, we define the *linear model*, or the *linear hypothesis class*, as the set of linear functions from the domain set to the response set:

$$\mathcal{H}_{reg} := \left\{ h(x_1,\ldots,x_d) = w_0 + \sum_{i=1}^{d} x_i w_i \;\middle|\; w_0, w_1, \ldots, w_d \in \mathbb{R} \right\}. \tag{1}$$

In statistics, learning f from a training sample is known as *linear regression*[1]. Each function $h \in \mathcal{H}_{reg}$ is characterized by the **weights** (also known as regression coefficients) $w_1, \ldots, w_d$ representing the d features and an **intercept** w_0. To simplify the notation, for a given sample $\mathbf{x} = (x_1, \ldots, x_d)^\top \in \mathbb{R}^d$ we add a zero-th coordinate with the value of 1, and define $\mathbf{x} = (1, x_1, \ldots, x_d)^\top \in \mathbb{R}^{d+1}$. Using this notation each function in the linear hypothesis class can be written in the form $h(\mathbf{x}) := \langle \mathbf{x}, \mathbf{w} \rangle = \mathbf{x}^\top \mathbf{w}$. For the remainder of this chapter, we will assume that the intercept is already incorporated into the weights vectors, so we can define linear hypothesis class equivalently as

$$\mathcal{H}_{reg} := \left\{ h_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^\top \mathbf{w} \mid \mathbf{w} \in \mathbb{R}^{d+1} \right\} \tag{2}$$

Note that by convention, the first coordinate of $\mathbf{w}$ is the intercept w_0.

So, given a training set S, we are looking for a vector $\mathbf{w} \in \mathbb{R}^{d+1}$ such that $y_i = \mathbf{x}_i^\top \mathbf{w}$ for all $i \in [m]$. This setup should be familiar from **??**. However, as we will see, we may not be able to find a vector $\mathbf{w}$ for which all these equalities hold exactly.

The regression matrix

Let us arrange the training data $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ in matrix form. We define the *response vector* as the column vector $\mathbf{y} \in \mathbb{R}^m$ and the *regression matrix* (or *design matrix*) $\mathbf{X} \in \mathbb{R}^{m \times (d+1)}$ as follows.

$$\mathbf{X} = \begin{bmatrix} — & \mathbf{x}_1 & — \\ — & \mathbf{x}_2 & — \\ & \vdots & \\ — & \mathbf{x}_m & — \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Note that m rows of $\mathbf{X}$ represent our m training samples and the $d+1$ columns of $\mathbf{X}$ represent the intercept and d features. In this notation, we are looking for a vector $\mathbf{w} \in \mathbb{R}^{d+1}$ that satisfies a system of m linear equations in the variable $\mathbf{w}$,

$$\mathbf{X}\mathbf{w} = \mathbf{y} \tag{3}$$

We must have enough training samples

At this point, we will assume that $m \geq d+1$, namely, that we have enough training samples so that the linear system (3) is not under-determined. In practical terms, this means that we have at least as many training samples as we have features. In our online store example, this means that we must collect data on $m \geq d+1$ customers before we start training our regression model, where d is the number of features we collect on each customer (e.g. age, income, total spending, number of monthly visits to the website, etc).

[1] The name “regression“ refers to a statistical phenomenon known as “regression to the mean”.

0.2 Designing A Learning Algorithm

0.2.1 Realizablity

Recall that to derive the problem of finding $\mathbf{w} \in \mathbb{R}^{d+1}$ that satisfies (3) we have restricted ourselves to describing functional relations $\mathcal{X} \xrightarrow{f} \mathcal{Y}$ such that $f \in \mathcal{H}_{reg}$. The case where there exists a solution for (3) is called the **Realizable** case. Let $\hat{\mathbf{w}}$ be a solution for (3), then the prediction rule we choose is $\hat{f}(\mathbf{x}) = \mathbf{x}^\top \hat{\mathbf{w}}$.

The case where there is no $f \in \mathcal{H}_{reg}$ that satisfies the system of equations (i.e there is no solution for the system) is called the **Non-Realizable** case. In this case, since we decided to choose a prediction rule in $\mathcal{H}_{reg}$, we must settle for finding $\hat{f} \in \mathcal{H}_{reg}$ which is *“most fitting“* for our purposes.

Our learning algorithm for linear regression must address both the realizable and non-realizable cases. In the realizable case, to find the rule f, all we need to do is solve the linear system (3) for $\mathbf{w}$. But what will we do in the non-realizable case, where $f \notin \mathcal{H}_{reg}$? How should we choose the prediction rule $\hat{f}$?

0.2.2 Loss Function

One way to choose $\hat{f} \in \mathcal{H}_{reg}$ in the non-realizable case is to assign each $f \in \mathcal{H}_{reg}$ with some measure of quality, and choose the "best" f. The function defined to measure the quality is called a **loss function** and it measures the quality of the hypothesis by comparing between the true- and predicted values:

$$\sum_{i=1}^{m} L\left(f(\mathbf{x}_i), \hat{f}(\mathbf{x}_i)\right), \quad i = 1, \ldots, m$$

We will then pick the prediction rule which is “best fitting“/“most likely“ given our training data and with respect to the loss function we chose. Two commonly used loss functions for regression problems are the **Absolute Value Loss**

$$L\left(y, \hat{f}(\mathbf{x})\right) := \left|y - \hat{f}(\mathbf{x})\right| \tag{4}$$

or the **Squared Loss**

$$L\left(y, \hat{f}(\mathbf{x})\right) := \left(y - \hat{f}(\mathbf{x})\right)^2 \tag{5}$$

We will focus on the linear regression setup when using the square loss function.

0.2.3 Empirical Risk Minimization

As we are concerned for the performance of a prediction rule $\hat{f}$ on a new data point $\mathbf{x}$ by $\left(\hat{f}(\mathbf{x}) - y\right)^2$, it makes sense to choose $\hat{f}$ that minimizes that same loss L *on the training data we already have*. This strategy for choosing $\hat{f}$ is known as **Empirical Risk Minimization**. For a given prediction rule $\hat{f} \in \mathcal{H}$, the quantity

$$\sum_{i=1}^{m} L\left(y_i, \hat{f}(\mathbf{x}_i)\right)$$

where $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ is our training data, is called the **empirical risk**. In the case of the square loss, the empirical risk of the linear function $\hat{f}(\mathbf{x}_i) = \mathbf{x}_i^\top \mathbf{w}$ is given by:

$$\sum_{i=1}^{m} \left(y_i - \mathbf{x}_i^\top \mathbf{w}\right)^2 = ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 = (\mathbf{y} - \mathbf{X}\mathbf{w})^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) \tag{6}$$

0.2.4 Least Squares

Minimizing the empirical risk of (6) means minimizing the sum of squares of the deviations of the responses from a linear function. In other words, we choose the linear function in $\mathcal{H}_{reg}$ that is closest to the responses in terms of the squared error distance. The deviation $y_i - \mathbf{x}_i^\top \mathbf{w}$ is called the i-th **residual** and the total empirical risk in our case is called **Residual Sum of Squares** (or **RSS**):

$$RSS_{\mathbf{X},\mathbf{y}}(\mathbf{w}) := ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2$$

To simplify notation we often write $RSS(\mathbf{w})$ keeping the dependence on $\mathbf{X}, \mathbf{y}$ implicit. So to learn the linear function by empirical Risk minimization we want to find

$$\underset{\mathbf{w}\in\mathbb{R}^d}{argmin}\, RSS(\mathbf{w}) = \underset{\mathbf{w}\in\mathbb{R}^d}{argmin}\, ||\mathbf{y} - \mathbf{X}\mathbf{w}||^2 \tag{7}$$

It is important to notice that the optimization problem (7) addresses both the realizable and non-realizable cases:

- In the realizable case, as $\mathbf{y} \in Im(\mathbf{X})$ we know there exists at least one solution $\widehat{\mathbf{w}}$ such that $\mathbf{X}\widehat{\mathbf{w}} = \mathbf{y}$. Such a solution will achieve a value of zero. As the RSS function is bounded below by zero, such a solution is therefore a minimizer of the RSS.
- In the non-realizable case, as $\mathbf{y} \notin Im(\mathbf{X})$ there is no solution $\widehat{\mathbf{w}}$ such that $\mathbf{X}\widehat{\mathbf{w}} = \mathbf{y}$. Therefore, no vector $\widehat{\mathbf{w}}$ will achieve a value of zero for the RSS objective. Instead, we decide to find a vector that is "good enough" in the sense of minimizing the squared loss.

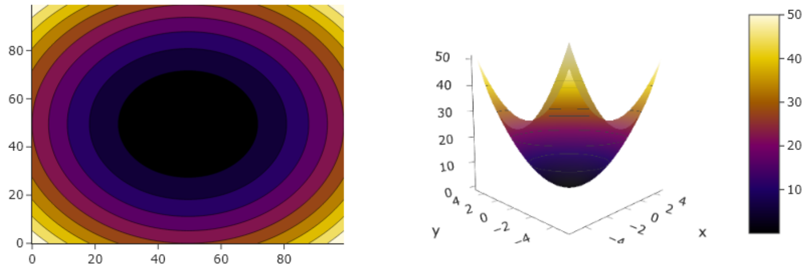

Figure 2: *Illustration of the RSS function over $\mathbb{R}^2$ for $\mathbf{X}$ of full rank.* *Chapter 2 Examples - Source Code*

A *necessary* condition for $\mathbf{w}$ to be a minimizer of the function $||\mathbf{y} - \mathbf{X}\mathbf{w}||^2$ is that all its partial derivative vanish at $\mathbf{w}$. Recalling the definition of the inner product, this condition can be written as:

$$\frac{\partial}{\partial w_j} RSS(\mathbf{w}) = -2\sum_{i=1}^{m} (\mathbf{x}_i)_j \cdot (y_i - \mathbf{x}_i \mathbf{w}) = 0 \tag{8}$$

for all $j = 0, \ldots, d$, where $(\mathbf{x}_i)_j$ is the j-th entry of $\mathbf{x}_i$. It is the $x_{j,i}$ element of the matrix $\mathbf{X}$. Notice that this constructs a system of $d+1$ linear equations in $\mathbf{w}$. We can organize (8) as such to get the form below. Recall that we have already derived this function in **??**.

$$\nabla RSS(\mathbf{w}) = -2\mathbf{X}^\top (\mathbf{y} - \mathbf{X}\mathbf{w}) = 0 \tag{9}$$

0.2.5 The Normal Equations

So a minimizer of (**??**) must also be a solution for the following linear system, known as the **Normal Equations**:

$$\mathbf{X}^\top(\mathbf{y}-\mathbf{X}\mathbf{w})=0 \quad\iff\quad \mathbf{X}^\top\mathbf{y}=\mathbf{X}^\top\mathbf{X}\mathbf{w} \tag{10}$$

Theorem 0.1 — The Non-Singular Case. Let $\mathbf{X}\in\mathbb{R}^{m\times(d+1)}$ with $m\geq d+1$. If $dim(Ker(\mathbf{X}))=0$ then $\widehat{\mathbf{w}}=\left[\mathbf{X}^\top\mathbf{X}\right]^{-1}\mathbf{X}^\top\mathbf{y}$ is a unique minimizer of (7).

Proof. As the kernel of $\mathbf{X}$ os trivial then $\mathbf{X}^\top\mathbf{X}$ has a trivial kernel. This means that the matrix $\mathbf{X}^\top\mathbf{X}$ is non-singular (i.e invertible) and $\left[\mathbf{X}^\top\mathbf{X}\right]^{-1}$ exists:

$$\begin{gathered}\mathbf{X}^\top\mathbf{y}=\mathbf{X}^\top\mathbf{X}\mathbf{w}\\ \Downarrow\\ \left[\mathbf{X}^\top\mathbf{X}\right]^{-1}\mathbf{X}^\top\mathbf{y}=\left[\mathbf{X}^\top\mathbf{X}\right]^{-1}\mathbf{X}^\top\mathbf{X}\mathbf{w}\\ \Downarrow\\ \widehat{\mathbf{w}}=\left[\mathbf{X}^\top\mathbf{X}\right]^{-1}\mathbf{X}^\top\mathbf{y}\end{gathered}$$

Lastly, we would like to show that $\mathbf{w}$, which we have found to be an extramum of the objective, is a minimum. Taking the second derivative with respect to the parameters:

$$\frac{\partial^2 RSS(\mathbf{w},\mathbf{X},\mathbf{y})}{\partial\mathbf{w}_k\partial\mathbf{w}_l}=\frac{\partial-2\sum_{i=1}^m\left(\mathbf{y}_i-\sum_{j=1}^d\mathbf{x}_{ij}\mathbf{w}_j\right)\mathbf{x}_k}{\partial\mathbf{w}_l}=2\sum_{i=1}^m\mathbf{x}_k\mathbf{x}_l=2\left[\mathbf{X}^\top\mathbf{X}\right]_{kl}\quad\forall k,l\in[d]$$

Notice that the matrix $\mathbf{X}^\top\mathbf{X}$ is a positive semi-definite matrix. Now, as we assumed that the columns of $\mathbf{X}$ are independent then for any $v\neq 0$ it holds that $v^\top\left[\mathbf{X}^\top\mathbf{X}\right]v=(\mathbf{X}v)^\top\mathbf{X}v=||\mathbf{X}v||^2>0$ and $\mathbf{X}^\top\mathbf{X}$ is a positive definite matrix. Thus, $\widehat{\mathbf{w}}$ is indeed a minima of the RSS as requested. ■

For the case where $m\geq d+1$ and $\mathbf{X}$ has a *non-trivial* kernel then there is an infinite number of solutions. Therefore, let us expand the definition of the inverse of a matrix.

Definition 0.1 Let $\mathbf{X}\in\mathbb{R}^{m\times(d+1)}$ and let $U\Sigma V^\top$ be its SVD. The **Moore-Penrose pseudoinverse** of $\mathbf{X}$ is $\mathbf{X}^\dagger=V\Sigma^\dagger U^\top$ where $\Sigma^\dagger$ is a $(d+1)\times m$ diagonal matrix defined by:

$$\Sigma_{i,i}^\dagger=\begin{cases}1/\Sigma_{i,i} & \Sigma_{i,i}\neq 0\\ 0 & \Sigma_{i,i}=0\end{cases}$$

Theorem 0.2 — Singular Case. Let $\mathbf{X}\in\mathbb{R}^{m\times(d+1)},\mathbf{y}\in\mathbb{R}^m$ be a regression problem where $m\geq d+1$. If $dim(ker(\mathbf{X}))\neq 0$ then:

$$\widehat{\mathbf{w}}:=X^\dagger\mathbf{y}$$

is a minimizer of (7).

Proof. Denote $r := rank(\mathbf{X})$ for which, since the kernel of $\mathbf{X}$ is non-trivial, $1 \leq r < d+1$ and $\sigma_1 \geq, \ldots, \geq \sigma_r > 0$. Let $\mathbf{X} = U\Sigma V^\top$ be the SVD of $\mathbf{X}$ so the columns of U, V provide orthonormal bases for the four fundamental subspaces:

$$\begin{array}{lrcl} U_{\mathcal{R}} \in \mathbb{R}^{m\times r} & \mathcal{R}(\mathbf{X}) & = & span\{\mathbf{u}_1, \ldots, \mathbf{u}_r\} \\ V_{\mathcal{R}} \in \mathbb{R}^{(d+1)\times r} & \mathcal{R}\left(\mathbf{X}^\top\right) & = & span\{\mathbf{v}_1, \ldots, \mathbf{v}_r\} \\ U_{\mathcal{N}} \in \mathbb{R}^{m\times(m-r)} & \mathcal{N}(\mathbf{X}) & = & span\{\mathbf{u}_{r+1}, \ldots, \mathbf{u}_m\} \\ V_{\mathcal{N}} \in \mathbb{R}^{(d+1)\times(d+1-r)} & \mathcal{N}\left(\mathbf{X}^\top\right) & = & span\{\mathbf{v}_{r+1}, \ldots, \mathbf{v}_{d+1}\} \end{array}$$

And $\mathcal{S} \in \mathbb{R}^{r\times r}$ the diagonal matrix with the r non negative singular values on its main diagonal: $\mathcal{S} := diag(\sigma_1, \ldots, \sigma_r)$. Using these notations, recall the compact SVD form of $\mathbf{X}$ **??**. So

$$\mathbf{X} := U\Sigma V^\top = \begin{bmatrix} U_{\mathcal{R}} & U_{\mathcal{N}} \end{bmatrix} \begin{bmatrix} \mathcal{S} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{\mathcal{R}}^\top \\ V_{\mathcal{N}}^\top \end{bmatrix} = U_{\mathcal{R}}\mathcal{S}V_{\mathcal{R}}^\top = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^\top$$

Now, for a minimizer of the RSS $\mathbf{w}$, from the normal equations:

$$\begin{array}{rcl} \mathbf{X}^\top\mathbf{y} & = & \mathbf{X}^\top\mathbf{X}\mathbf{w} \\ \widetilde{V}\widetilde{\Sigma}^\top\widetilde{U}^\top\mathbf{y} & = & \widetilde{V}\widetilde{\Sigma}^\top\widetilde{U}^\top\widetilde{U}\widetilde{\Sigma}\widetilde{V}^\top\mathbf{w} \\ \widetilde{\Sigma}\widetilde{U}^\top\mathbf{y} & = & \widetilde{\Sigma}^2\widetilde{V}^\top\mathbf{w} \end{array}$$

Since $\widetilde{\Sigma} \in \mathbb{R}^{r\times r}$ of full rank then Σ^{-1} exists and therefore:

$$\mathbf{w} = \widetilde{V}\widetilde{\Sigma}^{-1}\widetilde{U}^\top\mathbf{y}$$

Lastly, using the Moore-Perose pseudoinverse definition we "expand" the compact SVD form and conclude that:

$$\begin{array}{rcl} \widehat{w} & = & \widetilde{V}\widetilde{\Sigma}^{-}1\widetilde{U}^\top\mathbf{y} \\ & = & V\Sigma^\dagger U^\top\mathbf{y} \\ & = & \mathbf{X}^\dagger\mathbf{y} \end{array}$$

■

Another approach for finding the minimizer, is to express $||\mathbf{X}\mathbf{w} - \mathbf{y}||^2$ using the block forms of the range- and null spaces. It can be shown that

$$||\mathbf{X}\mathbf{w} - \mathbf{y}||^2 = \left|\left|\mathcal{S}V_{\mathcal{R}}^\top\mathbf{w} - U_{\mathcal{R}}^\top\mathbf{y}\right|\right|^2 + \left|\left|U_{\mathcal{N}}^\top\mathbf{y}\right|\right|^2$$

When searching for the minimizer only the first expression depends on $\mathbf{w}$ and we derive the same expression as derived above. However, when also looking at the second term we see that it in fact quantifies the portion of the data that resides in the null space and which we lose.

> **Corollary 0.3** $\widehat{\mathbf{w}} = \mathbf{X}^\dagger\mathbf{y}$ is always a solution of the Normal Equations

The estimator we found $\widehat{\mathbf{w}}$ is also referred to as the OLS estimator and is often noted as $\widehat{\mathbf{w}}^{OLS}$.

0.3 Categorical Variables

Consider the following problem. Suppose we want to predict the price of a house based on the following set of explanatory variables:

- The ‘House Size‘ is a numeric variable accepting positive numbers.
- The ‘Garden Size‘ is a categorical variable accepting the values: $small$, $medium$ and $large$.
- The ‘Number of Bedrooms‘ is a categorical numeric variable accepting natural numbers.
- The ‘House Type‘ is a categorical variable accepting the values: $private\,house$, $apartment$ and $studio-apartment$.

We would like to construct the following regression problem:

$$y = \mathbf{x}^\top \mathbf{w}, \quad \mathbf{w} = \begin{bmatrix} w_{\text{house size}} \\ w_{\text{garden size}} \\ w_{\text{number of bedrooms}} \\ w_{\text{house type}} \end{bmatrix}$$

Notice that we have different *types* of variables, and it is not clear how to treat each one of them. The ‘House Size‘ and ‘Number of Bedrooms‘ variables are *quantitative* variables where we have a natural order over the values. That is, we are able to state if one house is larger than another or if the number of bedrooms in a house is less than the number of bedrooms in another house. For these variables, solving the set of linear equations seen in a linear regression problem is something we know how to do.

In the case of the ‘Garden Size‘ variable, though the values are in fact not numeric ($small$, $medium$ and $large$), we are able to define a logical order over these values. It makes sense to state that $small < medium < large$ or that $large \neq small$. We can think of this order as some encoding of the non-numeric categories as numbers, with the order defined over these numbers being the order of the variable. For example: $small \mapsto 1, medium \mapsto 2, large \mapsto 3$.

Last, consider the ‘House Type‘ variable. Can we define some logical map as we did for the ‘Number of Bedrooms‘ variable? Are we able to state that in some sense that $studio-apartment > private-house$ or that $studio-apartment < private-house$? As we cannot find any logical ordering we must devise a different manner in which to deal with these variables. The most commonly used method is to encode these categories in what is known as *dummy variables* or *one-hot encoding*. Given a categorical variable with K categories we instead represent it as a binary vectors with K entries, where only one of these entries is “on“.

$$\mathbf{x}_{\text{house type}} = \text{‘}apartment\text{‘} \quad \Rightarrow \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \begin{array}{l} \text{house size} \\ \text{garden size} \\ \text{number of bedrooms} \\ \text{private-house} \\ \text{apartment} \\ \text{studio-apartment} \end{array}$$

where $x_4 + x_5 + x + 6 = 1$, $x_4, x_5, \mathbf{x}_6 \in \{0, 1\}$.

> Notice that by converting a single categorical variable with K categories to a one-hot encoding we are in-fact adding $K-1$ variables to our model. This addition has both influences on running times (as the algorithms we use are often polynomial in the number of features) and on the number of samples needed for learning.