# 1 Regression in general

### 1) Describe in no more than five sentences what regression in general means

Regression is a term used in a family of *supervised learning* models. This models have, among other features, a labeled dataset of $(x^i, y^i)$ pairs, from them it is assumed that they were generated by a $truefunction\ y^i = f(x^i)$, and a set of parameters, which have to be somehow adjusted and combined in the hypothesis function $h$ in order for it to approximate $f$.

In this context, regression refers to the technique of adjusting those parameters to the given dataset. If the domain of $y$ is discrete, the task is *classification*. It has subtypes depending on $h$ (for example, *linear regression* if $h$ is affine) and it is usually performed by minimizing the mean squared error between $y^i$ and $h(x^i)$ for all $x^i$ in the dataset.

### 2) What is the connection between regression and inference?

In logic, inference is the way/effect of reaching some conclusion that is truth based on some given assumptions. In inductive inference, there are given some observed instances (for example: an eucalyptus, a baobab), and, assuming that they have features in common, the process of infering leads to the definition of such features (for example, the definition of *tree*, or in the given example, even *all trees are perenne*).

This is pretty much analogous to the task described in 1) of learning the parameters that fit best to a given dataset, assuming some function $h$: in fact, the parameters are inferred. The only difference is that, for general inductive inference, the set doesn't have to be labeled (f.e., *clustering* is also a form of inference).

### 3) In what cases would you use logistic regression instead of linear regression?

As perfectly described in Russel&Norvig's *Artificial Intelligence: A Modern Approach*, page 725, classificating based on a threshold function presents several problems, which are all adressed by applying a logistic function (like the sigmoid) instead. So logistic regression is predominantly used in the task of *classification*. Of course, this doesn't exclude the use of logistic functions as basis for the "normal", continuous regression, but it is rare the case when it performs better than a multivariate linear regression, or using gaussian/polynomial kernels as basis functions.

### 4) What is the goal of the dual representation in logistic regression?

The goal is to make some calculations faster, or even possible: the dual representation basically states explicitly the equivalence between two spaces (for example, a vector space of reals and a vector space of some kernel function). Usually, some computations are much easier to perform in one of them, whereas the wanted result should apply to the other. Due to this equivalence, it is possible to switch the representation, perform the

computations, and then switch back the solutions. This is exactly what kernel machines do (see 8. for more details).

5) What are the risks in applying regression on a small (probably noisy) dataset, and how could you minimize it ?

When applying regression to any noisy dataset, the model has two main goals: neutralize the noise, and calculate the function that maximizes the likelihood. The general problem is that it is very difficult to know if the applied model is the most adequate: the smaller the dataset, more models will fit well to it. The model has to make also a prediction for the amount of noise: the smaller the dataset, the higher probability that this prediction will be wrong.
The only solution for this is to make assumptions on how does the *true function* look like, and incorporate them to the model. This can be done implicitly (like, for example, $L_2$ regularizing, choosing some specifical basis function, tweaking some *hyperparameters*) or explicitly (by stating some *prior distribution* and using *Bayes' Theorem* to infer the *posterior*. See 8. for more details on this).
The problem is, of course, that the model needs to make assumptions on the data it is trying to learn from: strong assumptions will make the few samples useless, and mild assumptions won't neutralize the uncertainties. The strategies to find an optimum are mainly two: *cross-validation* when maximizing the likelihood (which is very ineffective in small datasets, since they have to be split in, at least, 3 parts), or analyzing the *prior distribution* depending on those *hyperparameters* (which is only possible if the model is formuled explicitly, that is, in a Bayesian model).

6) What is the connection between Occams Razor, the curse of dimensionality, and sparse linear regression?

- Occam's Razor: dogma accepted by the scientifical discourse from which, among all valid explanations, the simplest is the most valid one.
- Curse of dimensionality: negative consideration about the fact that the size of a problem space grows exponentially with its amount of dimensions.
- Sparsity: a quality referring to the information density of a data representation. High sparsity means little information distributed in a big representation.

When the number of dimensions is much bigger than the number of samples, the dataset is very sparsely distributed in the problem space, wicht causes the negative effects described in 5) and referred to as the "Curse of dimensionality". Sparse linear regression neutralizes this effect by observing only a limited number of features in each "learning step" (for example by picking the features with non-zero values in each mini-batch of the stochastic gradient descent algorithm). This implies that features that are not observed most probably don't have an impact in the outcome, which is the bottom line of Occam's Razor.

7) What are the benefits of using a Bayesian Model for regression?

All advantages have been already mentioned in 5): Bayesian Models allow an explicit formulation of the prior assumptions, and the inference mechanisms. This helps in order to bring the highly empirical discourse of the current machine learning reasearch to a more structured frame, in which conclusions and assumptions can be more easily translated from one model to another. Plus, although it is computationally more expensive, such explicit formulation allows to find analytical expressions for the priors and their hyperparameters, which can help to find the desired values in a more meaningful way than cross-validation does.

8) What is the kernel trick?

The kernel trick is a concept related to the dual representation of a problem, explained in 4): in this context, there exists some *kernel function* such that it exists a bijection between their result space $K(\phi_K, x)$ and the vector space formed by the dot product between the parameter vector and the input vector $< \phi_R, x >$. This is not a trick *per se*, at least not if we refuse to rename the *Fourier Transform* to *Fourier Trick*. The positive meaning comes from the fact that many of this kernel functions are fast to compute, and result in a highly-dimensional, non-linear bijection to the vector space which would be otherwise very expensive to compute, or even impossible: the product of two *Radial Basis Functions (RBF)*, or gaussian kernels, is equivalent to a dot product in an infinite-dimensional space, since the taylor expansion of the exponential function is a polynom with infinite dimensions: $e^x = \sum_{k=0}^{\infty} \{\frac{x^k}{k!}\}$.

## 2 Logistic Regression – Application

*Suppose that the tests for the wine's attributes are that expensive that we can only chose 2 attributes to measure for classification, which attributes would you select?*

2. What I would do is, like in *Principal Component Analysis*, to perform an SVD decomposition of the design matrix$\Phi = U\Sigma V^*$, and pick the two columns with the highest corresponding entry in the $\Sigma$ diagonal matrix. This two components are the ones that "retain the maximal variance" of the data, wich means that, if we only take two attributes, those two are the ones that minimize the error between hypothesis and dataset among all other pairs.
This is so because $U$ and $V^*$ are pseudo-orthogonal matrices, which means that they don't alter the lengths or angle relations. So the amount of variance given by an attribute can be directly and efficiently readen in $\Sigma$. This way of linear compression is very efficient and useful in datasets with low variability (that is, a stable variance), but aren't good if the outliers of the data are important. In other words: the attribute with maximal variance is not always the attribute with the most important information.