Regularization Techniques

One way of avoiding overfitting is to use regularization methods. Let's illustrate it here with the **Ridge Regression** method. The basic idea is that we want to tell the parameters, the estimated parameters, not to be too much affected by noise. We want them to kind of behave themselves and kind of stay small so that they're not able to overfit the data.

How do we incentivize them?

We take the least squares criterion that we have, and we include in it, and a penalty term that penalizes the thetas if they are too large. So this tells the mathematics that we want to produce small thetas. It biases the thetas towards the direction of zero.

$$\min_{\theta}\left[\sum_{i=1}^{n}(Y_i-\theta^T\mathrm{X}_i)^2+\alpha\sum_{j=1}^{m}\theta_j^2\right]$$

$\alpha \geq 0$: regularization hyperparameter

On the other hand, what we gain from that bias is that the coefficients will be less prone to be affected directly by small noises and they will have a smaller variance. Here **alpha** is a constant. It is the **regularization parameter**, or actually, it's what's called the hyperparameter. Hyperparameter is like a knob that you can have in an algorithm to tune or tweak the algorithm.

For any given choice of alpha, we can do optimization and you get some coefficients, the optimal coefficients and you can try different choices of alpha and see which one works best. In this particular problem, computations are still easy because we're again minimizing a quadratic function of the thetas and so we can solve this problem by solving a system of linear equations, and that can be done instantly with today's software.

There's also an interpretation in this. The bias that we have, we are biasing the thetas towards zero and it has an interpretation in terms of Bayesian statistics. It's as if we're assuming that the thetas are random variables that have a prior distribution that’s centered around zero, and alpha has a relation to the variance of that prior distribution. So this is more of a curiosity point. But it's also interesting philosophically that the penalty term reflects or expresses biases, prior biases that we may have from the coefficients.

A very popular method for doing regularization and avoiding overfitting comes under the name of the **lasso**, which is sparsity enforcing in a sense. Lasso basically incentivizes the parameters to stay small, instead of fitting the noise, but does this in a special way. It includes a penalty in which the **absolute values of the thetas** are **penalized**. It tells them that I would really like them to be zero. So that introduces a bias in the algorithm. Of course, it biases the thetas to be closer to zero. Here **alpha** is a **regularization parameter**.

$$\min_{\theta} \left[\sum_{i=1}^{n} (Y_i - \theta^T \mathrm{X}_i)^2 + \alpha \sum_{j=1}^{m} |\theta_j| \right]$$

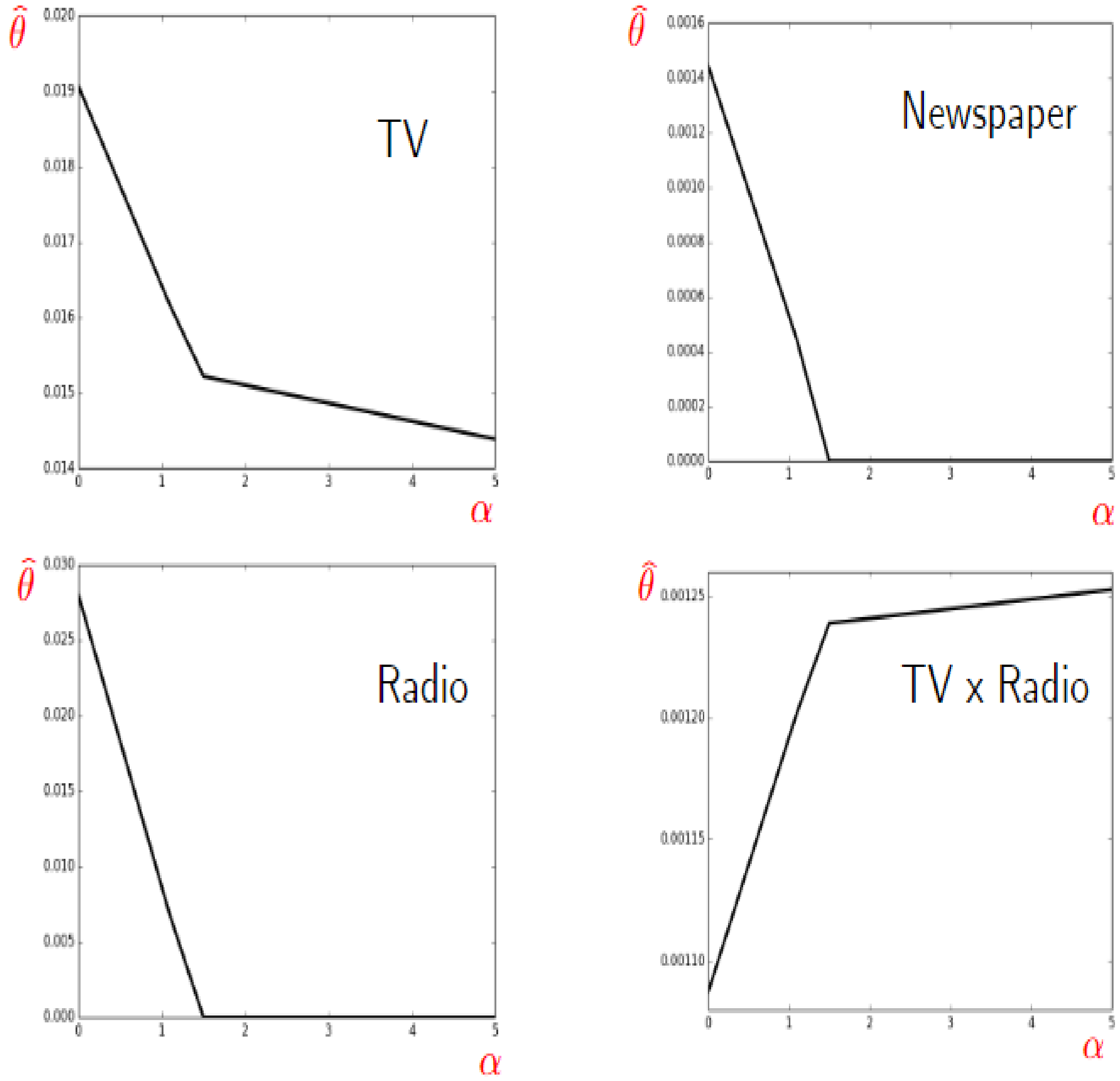
$\alpha \geq 0$: regularization hyperparameter

It's a non-negative parameter that you get to choose how many penalties you want to give on to the magnitudes of the coefficients. The way the algorithm is implemented in practice is that we try a few different values of alpha and we see which one is going to give us the best results. Now an interesting aspect of the optimal solutions to this regularized problem is that the optimal solution tends to set many of the theta j's to zero and so we get what's called a **sparse solution**.

Most of the components of theta are zeros and there are a few non-zeros. So this formulation underlies an assumption that we believe that most thetas are at zero, but maybe we do not know the locations of these zeros. When this problem is solved it is going to discover which ones are the zeros and which ones are the non-zeros.

The problem can be solved using efficient algorithms and it has been very much studied and has very strong theoretical guarantees. Under some assumptions, this formulation is going to discover the correct sparsity structure. If there is a true vector theta, which is sparse with lots of zeros, this formulation is going to discover where the zeros are, at least within the limit of a very large data set.

Let us consider the marketing example. When we run it with alpha equal to zero, the results that we get the coefficients are the ordinary regression coefficients that we have gotten before.



But as we change alpha, if we use a bigger alpha, this has the effect of moving the estimates closer to zero. Some estimates will move closer to zero but will not get to zero such as this **TV** for example, which survives, and this **TV * Radio** also survives. On the other hand some of the other coefficients, the **newspaper coefficient** gets driven to **zero**, and also the **radio coefficient** gets driven to **zero**.

When we put a big enough penalty on the thetas, what happens is that we end up with just two variables with TV and TV*radio. Now, if you were to push alpha even more and more, eventually everything would settle to zero.

And the question is, where in the middle, are we going to get the best possible results? But at least this figure doesn't tell us which one is the right value of alpha, but it does illustrate how increasing alpha tends to move the different parameters of different thetas towards zero.