Statistical Learning, Fall 2022

# Class notes 4

## Extensions of linear models and nearest neighbors in "modern" methods

The parametric model $\hat{y} = x^t\hat{\beta}$ seems quite limited, but we should note that it does not need to be linear in the original $p$ coordinates of $x$. For example, we can add all squares and products of the original coordinates, to get $h(x) \in \mathbb{R}^{p^2}$, and the model $\hat{y} = h(x)^t\hat{\beta}$ is now much richer, but since $h$ is a fixed transformation, it is a still linear in terms of estimating the parameters $\hat{\beta}$. In general, we can do a transformation $x \in \mathbb{R}^p \to h(x) \in \mathbb{R}^q$ with $q >> p$ (even $q = \infty$) to get linear models that are actually very rich. We will describe several modern methods (like boosting and kernel machines) in this context.

Taking it further, the coordinates of $h$ (which are functions themselves) can have learnable parameters, for example: $h_k(x) = \frac{1}{1+\exp(x^t\hat{\beta}_k)}$. This can now describe a neural network, in this case with sigmoid link.

We will also discuss modern methods in the context of non-parametric nearest-neighbor thinking, like thinking of trees (and random forests) as learning flexible neighborhoods/metrics. We may also discuss kernel methods as bridging some of the gap between the parametric and non-parametric thinking.

## Refresher on "classical" statistical methods and their connection to our setup

A typical statistical modeling approach often starts from assuming a *probabilistic* model that fits the data, with unknown parameters, and then using approaches like maximum likelihood estimation (MLE) to estimate the parameters from data. For example, we may *assume* that $y = x^t\beta + \epsilon$ , $\epsilon \sim N(0, \sigma^2)$. If we assume that and want to estimate $\beta$ using MLE from a training set $T = (\mathbb{X}, \mathbb{Y})$, we can write the log-likelihood (strictly speaking, this applies when we also assume Fixed-X, i.e. that $\mathbb{X}$ is not random, though it is almost accurate without this assumption as well):

$$\ell(\beta; \mathbb{Y}) = n \log\left(\frac{1}{\sqrt{2\pi}}\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - x_i^t\beta)^2 = const1 - const2 \times RSS(\beta),$$

hence regardless of the unknown $\sigma^2$, the MLE is not surprisingly the least squares estimator. If instead of assuming $\epsilon \sim N$ we assumed $\epsilon \sim Exp(\lambda)$ it is easy to show we would end up with

absolute regression as the MLE for $\beta$. Further, generalized linear models (GLM), like logistic regression for classification problems (which we will talk about) can also be interpreted in this fashion.

However, it is important to emphasize that the assumptions underlying the MLE approach (like true linear model or normality of errors) are ones we generally seek to avoid, and so when possible we prefer to treat the modeling approaches like least squares primarily as a black-box, whose properties we want to analyze in different settings — including but not limited to the ones that make them the MLE.

Another important point on classical results from a regression course, relates to the statistical inference that linear models and GLM offer, in particular the t-statistics for testing $H_{0j} : \beta_j = 0$, and ANOVA for selecting between models. These are hugely useful, but we must remember they are tightly tied to the probabilistic framework above, in particular they are valid if we assume:

1. Fixed-X
2. True linear model
3. $\epsilon \sim N(0, \sigma^2)$ normal i.i.d errors

In that respect we should enjoy the usefulness of these inferential tools, while remembering their limitations in an assumption-free situation.

## Regularization: in general and in linear regression

One way to think about fitting a predictive model: We have a large class of *candidate models* $\mathcal{F}$, and we want to choose a good one based on the training sample $T$, typically by minimizing some loss function:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} L(f; T).$$

Least squares regression follows this recipe exactly, with $\mathcal{F}$ being the family of all linear functions, and $L$ the RSS on the training data.

However, even in least squares regression, if $p$ is very big, in particular $p > n$, we may be able to find a perfect fit, which brings RSS to 0 (why is $p > n$ the relevant situation?). In cases where $\mathcal{F}$ is a bigger family — for example, all twice differentiable functions on $\mathbb{R}^p$ — it is guaranteed that many members of $\mathcal{F}$ will bring the RSS on $T$ to 0. Then it makes sense to add another criterion to choose between these functions, intuitively prefer "nice" (say, smooth) functions in $\mathcal{F}$ over others. Assume we have a measure of niceness $J(f)$. We can write the resulting problem as a constrained optimization problem:

$$\hat{f} = \arg\min_{f \in \mathcal{F}: L(f;T)=0} J(f) \tag{1}$$

A concrete example of this recipe is *smoothing splines interpolation* models: assume one dimensional data $x \in [0, 1]$, $\mathcal{F}$ is the family of all twice differentiable functions, and we choose $J(f)$ to be a natural smoothness criterion:

$$J(f) = \int_0^1 (f''(x))^2 dx.$$

So in problem (1) we are looking for the function that interpolates the data with the minimum lack of smoothness. This has a well known solution in the *interpolating cubic spline*. (The details are not critical here, only the general idea.)

More generally, we can relax the interpolation requirement that $L(f;T) = 0$ with a trade-off between loss and smoothness, and write in *penalized* form:

$$\hat{f}^{(pen)}(\lambda) = \arg\min_{f \in \mathcal{F}} L(f;T) + \lambda J(f),$$

or equivalently, in *constrained* form:

$$\hat{f}^{con}(s) = \arg \min_{f \in \mathcal{F}: J(f) \leq s} L(f;T).$$

These two formulations are equivalent in the sense that under pretty general conditions, $\forall \lambda \geq 0,\ \exists s \geq 0\ : \hat{f}^{(pen)}(\lambda) = \hat{f}^{(con)}(s)$.

For the penalized formulation, as $\lambda \to 0$ we converge to the interpolating solution above, and as $\lambda \to \infty$ we converge to the best solution with $J(f) = 0$. For the smoothing spline problem, where $L(f;T)$ is squared loss, and $J(f) = \int_0^1 (f''(x))^2 dx$, all the optimal solutions $\hat{f}(\lambda)$ are cubic splines, with changing degree of smoothness (see this example). In this case, linear functions have $J(f) = 0$ (why?), and hence as $\lambda \to \infty$, the solution $\hat{f}(\lambda)$ converges to the least squares solution.

Intuitively the role of regularization is to *decrease variance* by considering a subset of the class $\mathcal{F}$ (this is clearer in the constrained formulation), or preferring some functions to others. In the case of linear regression we will make this explicit and analyze it mathematically in various ways.

## Regularized linear regression

As we have shown, the variance of the least squares regression estimator is $\approx \sigma^2 p/n$. this may be small when $p$ is small but can be substantial in high-dimensional settings. This is the situation where it makes sense to regularize linear regression. Penalties used are most often the norms of the coefficient vector:

$$\hat{\beta}^{pen}(\lambda) = \arg\min_{\beta} RSS(\beta) + \lambda \|\beta\|_q^q\ ,\ \ \hat{\beta}^{con}(s) = \arg \min_{\beta: \|\beta\|_q^q \leq s} RSS(\beta).$$

The commonly used values of $q$ are 1 (lasso) and 2 (ridge), with $q = 0$ a special case of variable selection.

We can think of this geometrically as projecting the least squares solution on the constrained set, with a metric that depends on $\mathbb{X}$ (drawing on the board). We can also see from the geometry why not only the $\ell_0$ penalty but also the lasso penalty, tend to assign non-zero coefficients to only some of the variables. This additional variable selection can be useful both for improving prediction and for interpretation.

**Additional notes:**

- Typically in linear regression with $x \in \mathbb{R}^p$ we have $p+1$ variables, including the intercept $\hat{\beta}_0$. It is common practice not to include the intercept in the penalty, and have it non-penalized. This can complicate notations, so we do not separate it explicitly in our mathematical formulation (furthermore, if we assume that the columns are centered $\sum_i x_{ij} = 0 \forall j$, we can prove that the intercept can be fitted separately).
- If we rescale the columns of $\mathbb{X}$, then the least squares model is not affected, as we discussed in class. However, having different columns with different scale is problematic for penalty approaches, and so it is typical to standardize columns of $\mathbb{X}$ before applying regularization. However, like in k-NN, it may be a choice that depends on various considerations (like the "natural scale" of the variables).

**Variable/subset selection**

We can think of this approach as applying an $\ell_0$ penalty or simply as constraining the number of non-zero coefficients in the model:

$$\hat{\beta}^{(k)} = \arg \min_{|\{j:\beta_j != 0\}| \leq k} RSS(\beta).$$

Note that we have a total of $\binom{p}{k}$ such subsets, and so the complexity of solving this problem can become very high when $p, k$ are not small.

There are various heuristics for solving this approximately:

- Forward selection
- Backward elimination
- Forward-backward approaches which both add and remove variables

However it is generally not considered a favorable modern approach, compared to ridge and (especially) lasso. Specifically, ridge and lasso are *convex* formulations, and hence can be solved efficiently even in high dimension.

**Ridge regression**

Penalized ridge regression:

$$\hat{\beta}(\lambda) = \arg \min_{\beta} ||\mathbb{Y} - \mathbb{X}\beta||_2^2 + \lambda \|\beta\|_2^2.$$

Note that this is a quadratic function of the vector $\beta$, so we know we can solve it by differentiating and comparing to zero, giving the well known ridge regression solution:

$$-2X^t(Y - X\hat{\beta}(\lambda)) + 2\lambda\hat{\beta}(\lambda) = 0 \;\; \Rightarrow \hat{\beta}(\lambda) = (\mathbb{X}^t\mathbb{X} + \lambda I_p)^{-1}\mathbb{X}^t\mathbb{Y}.$$

We can see that the ridge solution is a slightly modified version of the least squares solution with the additional "ridge" on the diagonal inside the inverse. Note also that if $p > n$ then $\mathbb{X}^t\mathbb{X}$ is non invertible (rank $n$ at most), but $\mathbb{X}^t\mathbb{X} + \lambda I_p$ is guaranteed to be of full rank and invertible.

We are going to prove some interesting statistical results on ridge regression, including:

1. If the linear model is correct then, some ridge regression is always useful, that is the derivative of the prediction error at $\lambda = 0$ is negative (see HW2).

2. Compared to the least squares solution, the ridge regression solution with $\lambda > 0$ has lower prediction variance and higher prediction bias.

**Singular Value Decomposition (SVD) representation of ridge regression**

Recall the SVD: for $\mathbb{X}_{n\times p}$ with $n > p$ and rank $p$, we can always write it as:

$$\mathbb{X} = U_{n\times p} D_{p\times p} V^T_{p\times p},$$

where:

- $U_{n\times p}$ has orthonormal columns
- $D_{p\times p}$ is diagonal with non-negative entries $d_1 \geq ... \geq d_p \geq 0$ (the singular values, strictly positive if $X$ is rank $p$)
- $V_{p\times p}$ is an *orthogonal* matrix (meaning it has orthonormal rows and columns, $VV^T = V^TV = I$)

**Note:** The SVD above is the "economical" version. An alternative SVD is

$$\mathbb{X} = U_{n\times n} D_{n\times n} V^T_{n\times p},$$

where for example $D$ has only $p$ non-zero elements. However when $p > n$, this representation is actually the economical one...

Now we can use this to simplify(?) the writing of various expressions that are important for analyzing ridge regression:

- $\mathbb{X}^T\mathbb{X} = VDU^TUDV^T = VD^2V^T$
- $\left(\mathbb{X}^T\mathbb{X}\right)^{-1} = VD^{-2}V^T$ since $VD^{-2}V^TVD^2V^T = VD^{-2}D^2V^T = VV^T = I$ (where $D^{-2} = diag\left(d_j^{-2}\right)$)
- $\left(\mathbb{X}^T\mathbb{X} + \lambda I_p\right)^{-1} = \left(VD^2V^T + \lambda VIV^T\right)^{-1} = \left(V\left(D^2 + \lambda I\right)V^T\right)^{-1} = \left(V\left(D^2 + \lambda I\right)^{-1}V^T\right)$.
- $\mathbb{X}\left(\mathbb{X} + \lambda I\right)^{-1}\mathbb{X}^T = UD^2\left(D^2 + \lambda I\right)^{-1}U^T$.

Now we can use the SVD to analyze the variance of (Fixed-X) ridge regression:

$$\begin{aligned}
\textstyle\sum_{i=1}^n Var\left(\hat{y}_i(\lambda)\right) &= tr(Cov(\hat{Y}(\lambda))) = tr\left(Cov\left(\mathbb{X}\left(\mathbb{X} + \lambda I\right)^{-1}\mathbb{X}^T\mathbb{Y}\right)\right) = \\
&= \sigma^2 tr\left(UD^2\left(D^2 + \lambda I\right)^{-1}U^TU\left(D^2 + \lambda I\right)^{-1}D^2U^T\right) = \\
&= \sigma^2 tr\left(U^TUD^4\left(D^2 + \lambda I\right)^{-2}\right) = \sigma^2\sum_{i=1}^p \frac{d_j^4}{\left(d_j^2 + \lambda\right)^2}.
\end{aligned}$$

We see that this is a decreasing function of $\lambda$, with the value we already know at $\lambda = 0$:

$$\sigma^2 \sum_{i=1}^{p} \frac{d_j^4}{\left(d_j^2 + 0\right)^2} = p\sigma^2.$$

Similarly, we can assert that the squared bias is an increasing function of $\lambda$ (the proof is not as elegant). Note that:

$$\mathbb{E}\hat{Y} = \mathbb{E}\left[\mathbb{X}\left(\mathbb{X} + \lambda I\right)^{-1}\mathbb{X}^T\mathbb{Y}\right] = UD^2\left(D^2 + \lambda I\right)^{-1}U^T\mathbb{E}\mathbb{Y}.$$

Therefore:

$$\begin{aligned}\sum_i \left(\mathbb{E}(y_i) - \mathbb{E}(\hat{y}_i(\lambda))\right)^2 &= \left(\mathbb{E}\mathbb{Y} - \mathbb{E}\hat{Y}(\lambda)\right)^T\left(\mathbb{E}\mathbb{Y} - \mathbb{E}\hat{Y}(\lambda)\right) = \\ &= \mathbb{E}\mathbb{Y}^T U\left(I - 2diag\left(\frac{d_j^2}{d_j^2+\lambda}\right) + diag\left(\frac{d_j^4}{\left(d_j^2+\lambda\right)^2}\right)\right)U^T\mathbb{E}\mathbb{Y} := \mathbb{E}\mathbb{Y}^T M(\lambda)\mathbb{E}\mathbb{Y}.\end{aligned}$$

Now to prove that this is an increasing function of $\lambda$ we can differentiate it, equivalent to differentiating the inner matrix $M(\lambda)$, to get on the diagonal:

$$\frac{\partial}{\partial\lambda}\left(1 - 2\frac{d_j^2}{d_j^2+\lambda} + \frac{d_j^4}{\left(d_j^2+\lambda\right)^2}\right) = 2\frac{1}{d_j^2+\lambda}\left(\frac{d_j^2}{d_j^2+\lambda} - \left(\frac{d_j^2}{d_j^2+\lambda}\right)^2\right) > 0.$$

So we get that $\frac{\partial M(\lambda)}{\partial\lambda}$ is a positive definite matrix, which is the center of a quadratic in the full derivative, and therefore the squared bias is an increasing function of $\lambda$.

### Guaranteed error reduction from adding ridge

In the HW, you are asked to prove the following (which should be straight forward using the above formulas):
Assume:

- Fixed-X setting
- True linear model $\mathbb{E}\mathbb{Y} = \mathbb{X}\beta$.

Then the prediction error of ridge as a function of $\lambda$ has a negative derivative at $\lambda = 0$, and therefore adding a little ridge regularization is guaranteed to improve prediction!