EECS 189 Introduction to Machine Learning
Fall 2020 **Note 9**

1 Total Least Squares

Previously, we have covered **Ordinary Least Squares (OLS)** which assumes that the dependent variable y is noisy but the independent variables $\mathbf{x}$ are noise-free. We now discuss **Total Least Squares (TLS)**, where we assume that our independent variables are also corrupted by noise. For this reason, TLS is considered an **errors-in-variables** model.

1.1 A probabilistic motivation?

We might begin with a probabilistic formulation and fit the parameters via maximum likelihood estimation, as before. Consider for simplicity a one-dimensional linear model

$$y_{\text{true}} = wx_{\text{true}}$$

where the observations we receive are corrupted by Gaussian noise

$$(x, y) = (x_{\text{true}} + z_x, y_{\text{true}} + z_y) \qquad\qquad z_x, z_y \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$$

Combining the previous two relations, we obtain

$$\begin{aligned} y &= y_{\text{true}} + z_y \\ &= wx_{\text{true}} + z_y \\ &= w(x - z_x) + z_y \\ &= wx \underbrace{-wz_x + z_y}_{\sim\mathcal{N}(0,w^2+1)} \end{aligned}$$

The likelihood for a single point is then given by

$$P(x, y; w) = \frac{1}{\sqrt{2\pi(w^2 + 1)}} \exp\left(-\frac{1}{2}\frac{(y - wx)^2}{w^2 + 1}\right)$$

Thus the log likelihood is

$$\log P(x, y; a) = \text{constant} - \frac{1}{2}\log\left(w^2 + 1\right) - \frac{1}{2}\frac{(y - wx)^2}{w^2 + 1}$$

Observe that the parameter w shows up in three places, unlike the form that we are familiar with, where it only appears in the quadratic term. Our usual strategy of setting the derivative equal to zero to find a maximizer will not yield a nice system of linear equations in this case, so we'll try a different approach.

1.2 Low-rank formulation

To solve the TLS problem, we develop another formulation that can be solved using the singular value decomposition. To motivate this formulation, recall that in OLS we attempt to minimize $||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2$, which is equivalent to

$$\min_{\mathbf{w},\boldsymbol{\epsilon}} ||\boldsymbol{\epsilon}||_2^2 \quad \text{subject to} \quad \mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}$$

This only accounts for errors in the dependent variable, so for TLS we introduce a second residual $\boldsymbol{\epsilon}_{\mathbf{X}} \in \mathbb{R}^{n\times d}$ to account for independent variable error:

$$\min_{\mathbf{w},\boldsymbol{\epsilon}_{\mathbf{X}},\boldsymbol{\epsilon}_{\mathbf{y}}} \left\|\begin{bmatrix}\boldsymbol{\epsilon}_{\mathbf{X}} & \boldsymbol{\epsilon}_{\mathbf{y}}\end{bmatrix}\right\|_F^2 \quad \text{subject to} \quad (\mathbf{X} + \boldsymbol{\epsilon}_{\mathbf{X}})\mathbf{w} = \mathbf{y} + \boldsymbol{\epsilon}_{\mathbf{y}}$$

For comparison to the OLS case, note that the Frobenius norm is essentially the same as the 2-norm, just applied to the elements of a matrix rather than a vector.

From a probabilistic perspective, finding the most likely value of a Gaussian corresponds to minimizing the squared distance from the mean. Since we assume the noise is 0-centered, we want to minimize the sum of squares of each entry in the error matrix, which corresponds exactly to minimizing the Frobenius norm.

In order to separate out the terms being minimized, we rearrange the constraint equation as

$$\underbrace{\begin{bmatrix}\mathbf{X} + \boldsymbol{\epsilon}_{\mathbf{X}} & \mathbf{y} + \boldsymbol{\epsilon}_{\mathbf{y}}\end{bmatrix}}_{\in\mathbb{R}^{n\times(d+1)}}\begin{bmatrix}\mathbf{w} \\ -1\end{bmatrix} = \mathbf{0}$$

This expression tells us that the vector $\begin{bmatrix}\mathbf{w}^\top & -1\end{bmatrix}^\top$ lies in the nullspace of the matrix on the left. However, if the matrix is full rank, its nullspace contains only $\mathbf{0}$, and thus the equation cannot be satisfied (since the last component, -1, is always nonzero). Therefore we must choose the perturbations $\boldsymbol{\epsilon}_{\mathbf{X}}$ and $\boldsymbol{\epsilon}_{\mathbf{y}}$ in such a way that the matrix is not full rank.

It turns out that there is a mathematical result, the **Eckart-Young theorem**, that can help us pick these perturbations. This theorem essentially says that the best low-rank approximation (in terms of the Frobenius norm[1]) is obtained by throwing away the smallest singular values.

Theorem. *Suppose* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *has rank* $r \leq \min(m, n)$*, and let* $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top = \sum_{i=1}^r \sigma_i \mathbf{u}_i\mathbf{v}_i^\top$ *be its singular value decomposition. Then*

$$\mathbf{A}_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i\mathbf{v}_i^\top = \mathbf{U}\begin{bmatrix}\sigma_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & \cdots & 0 \\ 0 & 0 & \sigma_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0\end{bmatrix}\mathbf{V}^\top$$

where $k \leq r$*, is the best rank-k approximation to* $\mathbf{A}$ *in the sense that*

$$||\mathbf{A} - \mathbf{A}_k||_F \leq ||\mathbf{A} - \tilde{\mathbf{A}}||_F$$

for any $\tilde{\mathbf{A}}$ *such that* $\text{rank}(\tilde{\mathbf{A}}) \leq k$*.*

[1] There is a more general version that holds for any unitary invariant norm.

Let us assume that the data matrix $\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}$ is full rank.[2] Write its singular value decomposition:

$$\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix} = \sum_{i=1}^{d+1} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

Then the Eckart-Young theorem tells us that the best rank-d approximation to this matrix is

$$\begin{bmatrix}\mathbf{X} + \boldsymbol{\epsilon}_\mathbf{X} & \mathbf{y} + \boldsymbol{\epsilon}_\mathbf{y}\end{bmatrix} = \sum_{i=1}^{d} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

which is achieved by setting

$$\begin{bmatrix}\boldsymbol{\epsilon}_\mathbf{X} & \boldsymbol{\epsilon}_\mathbf{y}\end{bmatrix} = -\sigma_{d+1} \mathbf{u}_{d+1} \mathbf{v}_{d+1}^\top$$

The nullspace of our resulting matrix is then

$$\text{null}\left(\begin{bmatrix}\mathbf{X} + \boldsymbol{\epsilon}_\mathbf{X} & \mathbf{y} + \boldsymbol{\epsilon}_\mathbf{y}\end{bmatrix}\right) = \text{null}\left(\sum_{i=1}^{d} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top\right) = \text{span}\{\mathbf{v}_{d+1}\}$$

where the last equality holds because $\{\mathbf{v}_1, \ldots, \mathbf{v}_{d+1}\}$ form an orthogonal basis for $\mathbb{R}^{d+1}$. To get the weight $\mathbf{w}$, we find a scaling α such that $\begin{bmatrix}\mathbf{w}^\top & -1\end{bmatrix}^\top$ is in the nullspace, i.e.

$$\begin{bmatrix}\mathbf{w} \\ -1\end{bmatrix} = \alpha \mathbf{v}_{d+1}$$

Note that this requires the $(d+1)$st component of $\mathbf{v}_{d+1}$ to be nonzero. (See Section 1.3 for details.)

1.2.1 Noise, regularization, and reverse-regularization

In a sense, above we have solved the problem of total least squares by reducing it to computing an appropriate SVD. Once we have $\mathbf{v}_{d+1}$, or any scalar multiple of it, we simply rescale it so that the last component is -1, and then the first d components give us $\mathbf{w}$. However, we can look at this more closely to uncover the relationship between TLS and the ideas of regularization that we have seen earlier in the course.

Since $\mathbf{v}_{d+1}$ is a right-singular vector of $\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}$, it is an eigenvector of the matrix

$$\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}^\top \begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix} = \begin{bmatrix}\mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{y} \\ \mathbf{y}^\top\mathbf{X} & \mathbf{y}^\top\mathbf{y}\end{bmatrix}$$

So to find it we solve

$$\begin{bmatrix}\mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{y} \\ \mathbf{y}^\top\mathbf{X} & \mathbf{y}^\top\mathbf{y}\end{bmatrix}\begin{bmatrix}\mathbf{w} \\ -1\end{bmatrix} = \sigma_{d+1}^2 \begin{bmatrix}\mathbf{w} \\ -1\end{bmatrix}$$

From the top line we see that $\mathbf{w}$ satisfies

$$\mathbf{X}^\top\mathbf{X}\mathbf{w} - \mathbf{X}^\top\mathbf{y} = \sigma_{d+1}^2 \mathbf{w}$$

[2] This should be the case in practice because the noise will cause $\mathbf{y}$ not to lie in the columnspace of $\mathbf{X}$.

which can be rewritten as

$$(\mathbf{X}^\top\mathbf{X} - \sigma_{d+1}^2\mathbf{I})\mathbf{w} = \mathbf{X}^\top\mathbf{y}$$

Thus, assuming $\mathbf{X}^\top\mathbf{X} - \sigma_{d+1}^2\mathbf{I}$ is invertible (see the next section), we can solve for the weights as

$$\hat{\mathbf{w}}_{\text{TLS}} = (\mathbf{X}^\top\mathbf{X} - \sigma_{d+1}^2\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$$

This result is like ridge regression, but with a *negative* regularization constant!

Why does this make sense? One of the original motivations of ridge regression was to ensure that the matrix being inverted is in fact nonsingular, and subtracting a scalar multiple of the identity seems like a step in the opposite direction. We can make sense of this by recalling our original model:

$$\mathbf{X} = \mathbf{X}_{\text{true}} + \mathbf{Z}$$

where $\mathbf{X}_{\text{true}}$ are the actual values before noise corruption, and $\mathbf{Z}$ is a zero-mean noise term with i.i.d. entries. Then

$$\begin{aligned}
\mathbb{E}[\mathbf{X}^\top\mathbf{X}] &= \mathbb{E}[(\mathbf{X}_{\text{true}} + \mathbf{Z})^\top(\mathbf{X}_{\text{true}} + \mathbf{Z})] \\
&= \mathbb{E}[\mathbf{X}_{\text{true}}{}^\top\mathbf{X}_{\text{true}}] + \mathbb{E}[\mathbf{X}_{\text{true}}{}^\top\mathbf{Z}] + \mathbb{E}[\mathbf{Z}^\top\mathbf{X}_{\text{true}}] + \mathbb{E}[\mathbf{Z}^\top\mathbf{Z}] \\
&= \mathbf{X}_{\text{true}}{}^\top\mathbf{X}_{\text{true}} + \mathbf{X}_{\text{true}}{}^\top\underbrace{\mathbb{E}[\mathbf{Z}]}_{\mathbf{0}} + \underbrace{\mathbb{E}[\mathbf{Z}]^\top}_{\mathbf{0}}\mathbf{X}_{\text{true}} + \mathbb{E}[\mathbf{Z}^\top\mathbf{Z}] \\
&= \mathbf{X}_{\text{true}}{}^\top\mathbf{X}_{\text{true}} + \mathbb{E}[\mathbf{Z}^\top\mathbf{Z}]
\end{aligned}$$

Observe that the off-diagonal terms of $\mathbb{E}[\mathbf{Z}^\top\mathbf{Z}]$ terms are zero because the ith and jth rows of $\mathbf{Z}$ are independent for $i \neq j$, and the on-diagonal terms are essentially variances. Thus the $-\sigma_{d+1}^2\mathbf{I}$ term is there to compensate for the extra noise introduced by our assumptions regarding the independent variables.

For another perspective, note that

$$\mathbb{E}[\mathbf{X}^\top] = \mathbb{E}[(\mathbf{X}_{\text{true}} + \mathbf{Z})^\top] = \mathbb{E}[\mathbf{X}_{\text{true}}{}^\top + \mathbf{Z}^\top] = \mathbb{E}[\mathbf{X}_{\text{true}}{}^\top] + \mathbb{E}[\mathbf{Z}^\top] = \mathbf{X}_{\text{true}}{}^\top$$

If we plug this into the OLS solution (where we have assumed no noise in the independent variables), we see

$$\hat{\mathbf{w}}_{\text{OLS}} = (\mathbf{X}_{\text{true}}{}^\top\mathbf{X}_{\text{true}})^{-1}\mathbf{X}_{\text{true}}{}^\top\mathbf{y} = (\mathbb{E}[\mathbf{X}^\top\mathbf{X}] - \mathbb{E}[\mathbf{Z}^\top\mathbf{Z}])^{-1}\mathbb{E}[\mathbf{X}]^\top\mathbf{y}$$

which strongly resembles the TLS solution, but expressed in terms of expectations over the noise $\mathbf{Z}$.

So, is this all just a mathematical trick or is there a practical sense in which ridge regularization itself is related to adding noise? The math above suggests that we can take the original training data set and instead of working with that data set, just sample lots of points (say r times each) with i.i.d. zero-mean Gaussian noise with variance λ added to each of their features. Call this the $\mathbf{X}$ and have the corresponding $\mathbf{y}$ just keep the original y values. Then, doing ordinary least squares on this noisily degraded data set will end up behaving like ridge regression since the laws of large numbers will make $\frac{1}{r}\mathbf{X}^\top\mathbf{X}$ concentrate around $\mathbf{X}_{\text{true}}{}^\top\mathbf{X}_{\text{true}} + \lambda I$. Meanwhile $\mathbf{X}^\top\mathbf{y}$ will concentrate to $r\mathbf{X}_{\text{true}}{}^\top\mathbf{y}_{orig}$ with $O(\sqrt{r})$ noise on top of this by the Central Limit Theorem (if we used other-than-Gaussian noise to noisily resample), and straight variance-$O(r)$ Gaussian noise if we indeed used Gaussian noise. Putting them together means that the result of OLS with noisily augmented training data will result in approximately the same solution as ridge-regression, with the solutions approaching each other as the number of noisy copies r goes to infinity.

Why does this make intuitive sense? How can adding noise make learning more reliable? The intuitive reason is that this added noise destroys inadvertent conspiracies. Overfitting happens because the learning algorithm sees some degree of conspiracies between the observed training labels y and the input features. By adding lots of copies of the training data with additional noise added into them, many of these conspiracies will be masked by the added noise because they are fundamentally sensitive to small details — this is why they manifest as large weights $\mathbf{w}$. We know from our studies of the bias/variance tradeoff that having more training samples reduces this variance. Adding our own noisy samples exploits this variance reduction.

In many practical machine learning situations, appropriately adding noise to your training data can be an important tool in helping generalization performance.

1.3 Existence of the solution

In the discussion above, we have in some places made assumptions to move the derivation forward. These do not always hold, but we can provide sufficient conditions for the existence of a solution.

Proposition. *Let $\sigma_1, \dots, \sigma_{d+1}$ denote the singular values of $\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}$, and $\tilde{\sigma}_1, \dots, \tilde{\sigma}_d$ denote the singular values of $\mathbf{X}$. If $\sigma_{d+1} < \tilde{\sigma}_d$, then the total least squares problem has a solution, given by*

$$\hat{\mathbf{w}}_{\text{TLS}} = (\mathbf{X}^\top\mathbf{X} - \sigma_{d+1}^2\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}$$

Proof. Let $\sum_{i=1}^{d+1} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ be the SVD of $\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}$, and suppose $\sigma_{d+1} < \tilde{\sigma}_d$. We first show that the $(d+1)$st component of $\mathbf{v}_{d+1}$ is nonzero. To this end, suppose towards a contradiction that $\mathbf{v}_{d+1} = \begin{bmatrix}\mathbf{a}^\top & 0\end{bmatrix}^\top$ for some $\mathbf{a} \neq \mathbf{0}$. Since $\mathbf{v}_{d+1}$ is a right-singular vector of $\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}$, i.e. an eigenvector of $\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}^\top\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}$, we have

$$\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}^\top\begin{bmatrix}\mathbf{X} & \mathbf{y}\end{bmatrix}\begin{bmatrix}\mathbf{a}\\0\end{bmatrix} = \begin{bmatrix}\mathbf{X}^\top\mathbf{X} & \mathbf{X}^\top\mathbf{y}\\ \mathbf{y}^\top\mathbf{X} & \mathbf{y}^\top\mathbf{y}\end{bmatrix}\begin{bmatrix}\mathbf{a}\\0\end{bmatrix} = \sigma_{d+1}^2\begin{bmatrix}\mathbf{a}\\0\end{bmatrix}$$

Then

$$\mathbf{X}^\top\mathbf{X}\mathbf{a} = \sigma_{d+1}^2\mathbf{a}$$

i.e. $\mathbf{a}$ is an eigenvector of $\mathbf{X}^\top\mathbf{X}$ with eigenvalue σ_{d+1}^2. However, this contradicts the fact that

$$\tilde{\sigma}_d^2 = \lambda_{\min}(\mathbf{X}^\top\mathbf{X})$$

since we have assumed $\sigma_{d+1} < \tilde{\sigma}_d$. Therefore the $(d+1)$st component of $\mathbf{v}_{d+1}$ is nonzero, which guarantees the existence of a solution.

We have already derived the given expression for $\hat{\mathbf{w}}_{\text{TLS}}$, but it remains to show that the matrix $\mathbf{X}^\top\mathbf{X} - \sigma_{d+1}^2\mathbf{I}$ is invertible. This is fairly immediate from the assumption that $\sigma_{d+1} < \tilde{\sigma}_d$, since this implies

$$\sigma_{d+1}^2 < \tilde{\sigma}_d^2 = \lambda_{\min}(\mathbf{X}^\top\mathbf{X})$$

giving

$$\lambda_{\min}(\mathbf{X}^\top\mathbf{X} - \sigma_{d+1}^2\mathbf{I}) = \lambda_{\min}(\mathbf{X}^\top\mathbf{X}) - \sigma_{d+1}^2 > 0$$

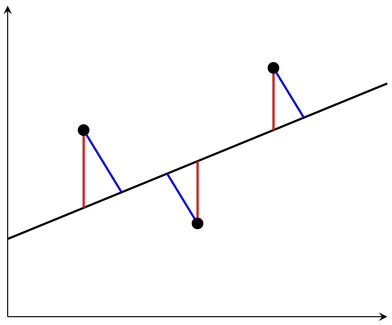
which guarantees that the matrix is invertible. □

This gives us a nice mathematical characterization of the existence of a solution, showing that the two technical requirements we raised earlier (the last entry of $\mathbf{v}_{d+1}$ being nonzero, and the matrix $\mathbf{X}^\top\mathbf{X} - \sigma_{d+1}^2$ being invertible) happen together. However, is the assumption of the proof likely to hold in practice? We give an intuitive argument that it is.

Consider that in solving the TLS problem, we have determined the error term $\boldsymbol{\epsilon}_{\mathbf{X}}$. In principle, we could use this to denoise $\mathbf{X}$, as in $\hat{\mathbf{X}}_{\text{true}} = \mathbf{X} - \boldsymbol{\epsilon}_{\mathbf{X}}$, and then perform OLS as normal. This process is essentially the same as TLS if we compare the original formulations. Assuming the error is drawn from a continuous distribution, the probability that the denoised matrix $\hat{\mathbf{X}}_{\text{true}}$ has collinear columns is zero.

1.4 TLS minimizes perpendicular distance

Recall that OLS tries to minimize the vertical distance between the fitted line and data points. TLS, on the other hand, tries to minimize the perpendicular distance. For this reason, TLS may sometimes be referred to as **orthogonal regression**.



The red lines represent vertical distance, which OLS aims to minimize. The blue lines represent perpendicular distance, which TLS aims to minimize. Note that all blue lines are perpendicular to the black line (hypothesis model), while all red lines are perpendicular to the x axis.