CSE 392: Matrix and Tensor Algorithms for Data

Instructor: Shashanka Ubaru

University of Texas, Austin

Spring 2024

Lecture 3: Least squares regression and kernel methods

Outline

Data fitting - Regression

- We are given,
  - A data matrix $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ with n samples $\{\boldsymbol{a}_i\}_{i=1}^n \in \mathbb{R}^d$ of d-dimensional features, and
  - A column vector $\boldsymbol{b} \in \mathbb{R}^n$ (targets).
- **Data fitting:** Find a functional relation between features and targets wrt. certain loss. General form: For a loss function $\ell(\cdot, \cdot)$, and a function $f(\cdot, \theta)$, where θ are the function parameters over a possible set Θ, we solve

$$\theta^* = \min_{\theta \in \Theta} \sum_{i=1}^{n} \ell(f(\boldsymbol{a}_i, \theta), b_i)$$

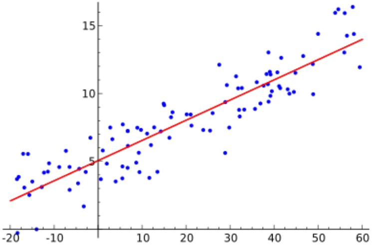
- **Numerous applications** from scientific computing to machine learning, finance, statistics and many more.

Least squares linear regression

- In the *least-squares* regression problem, assuming $d < n$, we solve:

$$\boldsymbol{x}^* = \min_{\boldsymbol{x} \in \mathbb{R}^d} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2.$$

- A linear function and Euclidean- (ℓ_2) norm (squared) loss function.
- The observed targets, $b_i = \boldsymbol{a}^\top \boldsymbol{x} + \varepsilon_i$, for $i = 1, \ldots, n$ and ε_i is noise..



Normal equation

The vector $\boldsymbol{x}^*$ minimizes $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2$ if and only if it is the solution of the **normal equations**:

$$\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}.$$

Normal equation

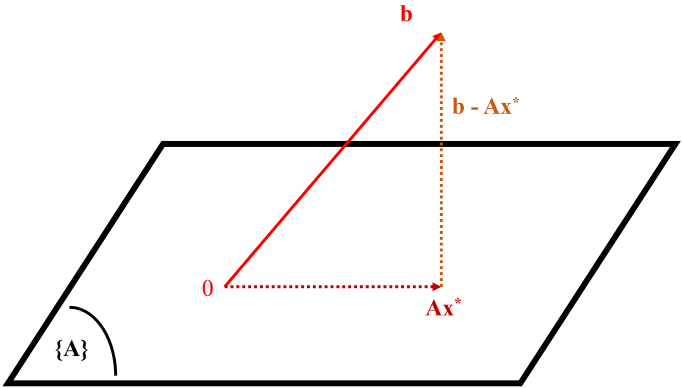
The vector $\boldsymbol{x}^*$ minimizes $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2$ if and only if it is the solution of the **normal equations**:

$$\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}.$$

Proof: Consider any $\tilde{\boldsymbol{x}} = \boldsymbol{x}^* + \Delta\boldsymbol{x}$, then we have

$$\begin{aligned}
\|\boldsymbol{A}\tilde{\boldsymbol{x}} - \boldsymbol{b}\|^2 &= \|\boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{A}\Delta\boldsymbol{x} - \boldsymbol{b}\|^2 \\
&= \|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|^2 - 2(\boldsymbol{A}\Delta\boldsymbol{x})^\top(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}) + \|\boldsymbol{A}\Delta\boldsymbol{x}\|^2 \\
&= \|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|^2 - 2(\Delta\boldsymbol{x})^\top \underbrace{\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b})}_{\nabla_{\boldsymbol{x}}\ell} + \underbrace{\|\boldsymbol{A}\Delta\boldsymbol{x}\|^2}_{\geq 0}
\end{aligned}$$

Hence, $\|\boldsymbol{A}(\boldsymbol{x}^* + \Delta\boldsymbol{x}) - \boldsymbol{b}\|^2 \geq \|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|^2$ for any $\Delta\boldsymbol{x}$, iff the gradient vector $\nabla_{\boldsymbol{x}}\ell$ is zero.



$\boldsymbol{x}^*$ is the best approximation to $\boldsymbol{b}$ from the subspace span$\{\boldsymbol{A}\}$ iff $(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x})$ is $\perp$ to the whole subspace span$\{\boldsymbol{A}\}$. This in turn is equivalent to Normal equations

$$\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}) = 0.$$

Matlab demo

Issue with normal equations

The solution is $\boldsymbol{x}^* = (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \boldsymbol{b}$.

- **Condition number** of a matrix :

$$\kappa_2(\boldsymbol{A}) = \|\boldsymbol{A}\|_2 \|\boldsymbol{A}^{-1}\|_2 = \sigma_{\max} / \sigma_{\min}$$

- Then, $\kappa_2(\boldsymbol{A}^\top \boldsymbol{A}) = \|\boldsymbol{A}^\top \boldsymbol{A}\|_2 \|(\boldsymbol{A}^\top \boldsymbol{A})^{-1}\|_2 = (\sigma_{\max} / \sigma_{\min})^2$.

E.g., suppose we have a matrix with spectrum in $[1, \epsilon]$, i..e, $\kappa_2(\boldsymbol{A}) = 1/\epsilon$.
Then, $\kappa_2(\boldsymbol{A}^\top \boldsymbol{A}) = \epsilon^{-2}$.
$\boldsymbol{A}^\top \boldsymbol{A}$ could be highly *ill-conditioned*.

Ridge Regression

Ridge Regression or Tikhonov regularization: For a given $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{b} \in \mathbb{R}^n$ the ridge-regression estimator is the minimizer of the problem:

$$\boldsymbol{x}_{rr} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2,$$

where $\lambda > 0$ is a fixed regularization parameter.

The solution is $\boldsymbol{x}_{rr} = (\boldsymbol{A}^\top\boldsymbol{A} + \lambda\boldsymbol{I})^{-1}\boldsymbol{A}^\top\boldsymbol{b}$.

We select an appropriate λ such that:

- we have a better conditioned matrix, and
- we avoid *over fitting*.

Bias–variance tradeoff.

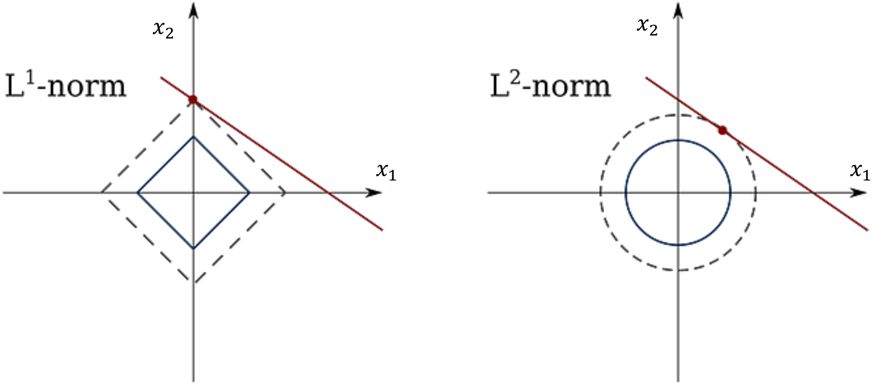
LASSO Regression

Least absolute shrinkage and selection operator, or LASSO , proposed by Thibshirani in 1996, solves the optimization problem:

$$\boldsymbol{x}_{lasso} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_1,$$

where $\lambda > 0$ is a fixed regularization parameter.

- The problem is still convex, but is non-smooth.
- Many efficient optimization algorithms have been proposed. E.g., Fast Iterative Shrinkage-Thresholding Algorithm (FISTA), Alternating Direction Method of Multipliers (ADMM).
- Yields a *sparse solution.*



Constraint Regions for LASSO (left) and Ridge Regression (right). Shows why LASSO yields a sparse solution.

Matlab demo II

Feature maps

- Linear regression fits a linear functions to the data.
- However, the functional relation could be "non-linear".
- **Example:** Consider fitting a cubic function:

$$b = x_3 a^3 + x_2 a^2 + x_1 a + x_0.$$

- We can view the cubic function as a **linear function** over a different set of feature variables. Let the function $\phi : \mathbb{R} \to \mathbb{R}^4$ be defined as:

$$\phi(a) = [1;\ a;\ a^2;\ a^3].$$

- If $\boldsymbol{x} = [x_0, x_1, x_2, x_3]$, then

$$b = x_3 a^3 + x_2 a^2 + x_1 a + x_0 = \boldsymbol{x}^\top \phi(a).$$

- The function ϕ is called the **feature map**.

Kernelization

- Approach to linearize non-linear problems.
- Map rows of $\boldsymbol{A}$ to $\phi(\boldsymbol{a}_i)$ in *higher dimension*.
- **Kernel Trick** or kernel substitution: if the input enters an algorithm only in the form of inner products, then we can replace the inner product with some other choice of a kernel.
- **Kernel:** corresponding to the feature map ϕ satisfies:

$$K(\boldsymbol{a}, \tilde{\boldsymbol{a}}) = \phi(\boldsymbol{a})^\top \phi(\tilde{\boldsymbol{a}})$$

- Kernel is symmetric of its arguments , i.e., $K(\boldsymbol{a}, \tilde{\boldsymbol{a}}) = K(\tilde{\boldsymbol{a}}, \boldsymbol{a})$.

Kernel properties

Mercer Theorem

Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be given. Then for K to be a valid (Mercer) kernel, it is necessary and sufficient that for any $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}, (n < \infty)$, the corresponding kernel matrix is symmetric positive semi-definite.

Kernel properties

Mercer Theorem

Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be given. Then for K to be a valid (Mercer) kernel, it is necessary and sufficient that for any $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}, (n < \infty)$, the corresponding kernel matrix is symmetric positive semi-definite.

Proof: Let the kernel matrix $\boldsymbol{K}$ be defined as $K_{ij} = \phi(\boldsymbol{a}_i)^\top \phi(\boldsymbol{a}_j)$. If K is a valid kernel, then $K_{ij} = \phi(\boldsymbol{a}_i)^\top \phi(\boldsymbol{a}_j) = \phi(\boldsymbol{a}_j)^\top \phi(\boldsymbol{a}_i) = K_{ji}$, hence symmetric. Also for any vector $\boldsymbol{z}$, we have:

$$
\begin{aligned}
\boldsymbol{z}^\top \boldsymbol{K} \boldsymbol{z} &= \sum_i \sum_j z_i K_{ij} z_j = \sum_i \sum_j z_i \phi(\boldsymbol{a}_i)^\top \phi(\boldsymbol{a}_j) z_j \\
&= \sum_i \sum_j z_i \sum_k \phi_k(\boldsymbol{a}_i) \phi_k(\boldsymbol{a}_j) z_j = \sum_k \sum_i \sum_j z_i \phi_k(\boldsymbol{a}_i) \phi_k(\boldsymbol{a}_j) z_j \\
&= \sum_k \left(\sum_i z_i \phi_k(\boldsymbol{a}_i) \right)^2 \geq 0.
\end{aligned}
$$

Kernels as similarity metrics

- Intuitively, when $\phi(\boldsymbol{a})$ and $\phi(\tilde{\boldsymbol{a}})$ are close to each other, the kernel $K(\boldsymbol{a}, \tilde{\boldsymbol{a}}) = \phi(\boldsymbol{a})^\top \phi(\tilde{\boldsymbol{a}})$ should be large.
- Conversely, if they are far apart, $K(\boldsymbol{a}, \tilde{\boldsymbol{a}})$ should be small.
- Kernel as a similarity measure of the features.
- **Gaussian Kernel:** Homogeneous kernels defined by the magnitude of distance:

$$K(\boldsymbol{a}, \tilde{\boldsymbol{a}}) = \exp\left(-\frac{\|\boldsymbol{a} - \tilde{\boldsymbol{a}}\|}{2\sigma^2}\right).$$

  It corresponds to an infinite dimensional feature map ϕ.

Kernel Ridge Regression

- Kernel methods - do not explicitly define or compute the feature map ϕ. Only compute the kernel function $K(\cdot, \cdot)$.
- In ridge regression, suppose we replace the feature vectors: $\boldsymbol{a}_i \to \Phi_i = \phi(\boldsymbol{a}_i)$ to account for non-linear function relation.
- Now the dimension can be much higher.
- The solution to the ridge regression is, with $\phi(\boldsymbol{a}_i)$'s as columns of Φ :

$$\boldsymbol{x}_{kr} = (\Phi\Phi^\top + \lambda \boldsymbol{I})^{-1}\Phi \boldsymbol{b} = \Phi\ (\Phi^\top\Phi + \lambda \boldsymbol{I})^{-1}\boldsymbol{b}$$

- Given a new data point $\boldsymbol{a}$, the prediction will be:

$$b = \phi(\boldsymbol{a})^\top \boldsymbol{x}_{kr}\ = \phi(\boldsymbol{a})^\top \Phi\ (\Phi^\top\Phi + \lambda \boldsymbol{I})^{-1}\boldsymbol{b} = \kappa(\boldsymbol{a})(\boldsymbol{K} + \lambda \boldsymbol{I})^{-1}\boldsymbol{b},$$

where $\kappa(\boldsymbol{a}) = [K(\boldsymbol{a}_i, \boldsymbol{a})]_{i=1}^n$.

Questions?