Machine Learning Foundations
Week 4

Shubharup G.
BSDS, IIT Madras

October 2023

Contents

1 Linear Regression

We believe that the value of a target variable is determined in major part as a linear combination aka weighted sum of the features.

The aim is to have the model learn this relationship viz. learn the weights (aka parameters) characterizing this relationship.

1.1 Linear Regression with SSE

Sum of Squared Error (SSE) is a loss function that is equal to the sum of squared errors i.e. the sum of the squared difference between the predicted output $\hat{y}_j$ and the actual output y_j, summed over the entire training dataset i.e. $\forall j$.

$$\boxed{\mathcal{L}(w) = \sum_{j=1}^{n}(\hat{y_j} - y_j)^2 = \sum_{j=1}^{n}({x_j}^T w - y_j)^2}$$

Alternatively, we can write $\mathcal{L}(w)$ in vector notation:

$$\boxed{\mathcal{L}(w) = ||\hat{y} - y||^2 = ||Xw - y||^2}$$

where X contains the **datapoints stacked as rows**.

The goal is to learn the optimal weights w^* which minimizes $\mathcal{L}(w)$ i.e.:

$$\boxed{w^* = argmin_w \mathcal{L}(w)}$$

We know from our earlier course in calculus that, when we want to minimize (or maximize) a function, we set the derivative equal to 0. Here, the loss function $\mathcal{L}(w)$ is a function of a vector of parameters w, so we set the *gradient* (vector of partial derivatives) equal to 0. Moreover, the loss function is a sum of squares, which is known to be convex (U-shaped or like a valley); so the stationary point we find will be the minima (rather than the maxima or a saddle).

$$\boxed{\nabla_w \mathcal{L}(w) = 0 \implies \nabla_w ||Xw - y||^2 = 0}$$

After some algebraic simplification, we get:

$$\boxed{(X^T X)w = X^T y}$$

We can further write that:

$$\boxed{w = (X^T X)^{-1} X^T y}$$ [1]

if $(X^T X)$ is full rank and thus invertible. Let $X \in \mathbb{M}_{n \times m}$ i.e. X as n datapoints each with m features. Then $(X^T X) \in \mathbb{M}_{m \times m}$ is invertible iff. $m \leq n$ (more datapoints than features) and $rank(X) = m$ (set of features are independent).

[1] $A^\dagger = (A^T A)^{-1} A^T$ is the Moore-Penrose pseudo-inverse. $\tilde{x} = A^\dagger b$ is the vector that comes closest to solving $Ax = b$ even when A is not invertible, for whatever reason: either it is square but rank-deficient, or it is rectangular. You can verify that the pseudo-inverse degenerates to the normal inverse, $A^\dagger = A^-$ when A *is* invertible. See Moore–Penrose inverse (Wikipedia).

1.2 Linear Regression as MLE

Maximum Likelihood Estimation is a parameter estimation technique, in which we choose that set of parameter values which maximizes the joint likelihood of seeing the set of observations given the assumption of the observations coming from a particular distribution.

It can be shown that **assuming Gaussian distribution for the noise** in the process which generates the labels, gives the same w^* as **choosing SSE as loss** function in Linear Regression.

$$\boxed{\textbf{assuming Gaussian noise in label generation} \implies \textbf{Choosing SSE as loss in LR}}$$

The main idea is that the observations for the target variable viz. the y_j's have two components — a stable component, and some noise. This stable component is stable as in it is rule-governed: it is a linear combination aka a weighted sum of the respective feature values. However, when we record observations of the target variable in the real world, there is generally some noise contained in them, either from invisible factors and/or factors beyond our control.

$$\boxed{y_j = \underbrace{< w, x_j >}_{\text{stable component}} + \underbrace{\epsilon}_{\text{noise}}}$$

Now if you assume that the noise ϵ comes from a zero-mean Gaussian:

$$\boxed{\epsilon \sim \mathcal{N}(0, \sigma^2)}$$

then each observation of the target variable itself came from that same Gaussian, only with its mean shifted:

$$\boxed{y_j | x_j \sim \mathcal{N}(w^T x_j, \sigma^2)}$$

Now if you calculated w^* which maximized the joint likelihood of seeing the observations:

$$\boxed{w^* = argmax_w \Pi_{i=1}^n \mathcal{L}_(y_i|x_i) = argmax_w \Pi_{i=1}^n \mathcal{N}(w^T x_i, \sigma^2, y_i)}$$

then this w^* would be the same as what you would get in Linear Regression by minimizing SSE.

1.3 Polynomial Regression

Each datapoint is "expanded". For illustration, if each datapoint has 1 feature only:

$$\boxed{x_j = \{f_{j1}\} \longrightarrow x_j = \{1, f_{j1}, f_{j1}^2, f_{j1}^3 \dots f_{j1}^z\}}$$

This expanded representation of x_j is denoted as $\phi(x_j)$. The matrix containing these expanded datapoints is denoted as $\Phi(X)$.[2]

Everything else is the same. Linear regression is run over $\Phi(X)$ as the data matrix. Similar to Section 1.1:

$$\boxed{w = (\Phi(X)^T \Phi(X))^{-1} \Phi(X)^T y}$$

[2] An instance of the Vandermonde matrix in form. See Vandermonde matrix (Wikipedia).

Worked out example What is the best fit line for the data below?

x	y
0	0
1.5	1.5
4	1

$$\Phi(X) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1.5 & 2.25 \\ 1 & 4 & 16 \end{bmatrix} \text{ and } y = \begin{bmatrix} 0 \\ 1.5 \\ 1 \end{bmatrix}$$

So $w = (\Phi(X)^T\Phi(X))^{-1}\Phi(X)^T y = \begin{bmatrix} -1.37 \\ 1.45 \\ -3.00 \end{bmatrix}$

So the best-fit line is $\hat{y} = (-1.37) + 1.45x - 3x^2$

1.4 Regularization and Ridge Regression

1.4.1 Regularization

Regularization is an essential technique / concept in Machine Learning, and has two purposes:

1. To prevent the model from over-fitting, because over-fitting leads to poor model performance on unseen data (aka poor generalization).
2. To impose some desired characteristic(s) on the solution set of parameter values to be learned.

1.4.2 Ridge Regression

In Ridge Regression, we minimize a modified loss function:

$$\widetilde{\mathcal{L}}(w) = \underbrace{\mathcal{L}(w)}_{\text{SSE / Original LR loss}} + \underbrace{||w||^2}_{\text{Regularizer}}$$

This particular Regularizer is called the **L2 regularizer**. Adding an L2 regularizer to the objective function incentivizes the learned optimal weights to be as small as possible while still minimizing the SSE loss.

Additionally, with the L2 regularization in Ridge Regression, **smaller weights distributed over all features (a symmetric distribution) is preferred** to large weights on some features and small weights on others (an asymmetric distribution). This contrasts with something called LASSO regression which has L1 Regularization i.e. where the Regularization term is $||w||$ (L1 norm aka Manhattan norm or "taxicab" norm) instead of $||w||^2$ (L2 norm aka Euclidean norm): L1 regularization also incentivizes small weights but "knocks" off unimportant features by giving them zero weight.

The general form for L2 or L1 regularization has a penalty term:

$$\widetilde{\mathcal{L}}(w) = \mathcal{L}(w) + \underbrace{\lambda}_{\text{reg. penalty}} ||w||^2$$

The regularization penalty λ controls how strongly we want to disincentivize the weights from growing large: higher lambda, stronger disincentive. If we set $\lambda = 0$ i.e. no penalty, we get the normal unregularized linear regression solution.

It turns out that Ridge Regression (but not LASSO) has a closed form expression for the optimal weights similar to unregularized linear regression:

$$w_{reg} = (X^T X + \lambda I)^{-1} X^T y$$

2 Eigenvectors and Eigenvalues

For a matrix A, its eigenvectors are special vectors for which the matrix acts like a scalar.

Motivation

There are many ways to motivate eigenvectors: one of them is **reduced computational complexity** of matrix multiplication.

If a matrix has enough number of linearly independent eigenvectors, then we can collect those eigenvectors into a basis for the input space of the matrix, appropriately called an **eigenbasis**. Subsequently, applying A to any vector in the input space can be done much faster if we first compute the representation of the vector in terms of the eigenbasis eigenvectors.

So whereas matrix multiplication is very computationally expensive, with eigenvectors it is possible to replace a bruteforce matrix multiplication with a smart scalar multiplication. One application of this explored in a later lecture of this course (and correspondingly a later section of this document) is using eigenvectors to directly and inexpensively approximate the n^{th} term in the Fibonnaci sequence.

2.1 Vectors and matrices over a field

Matrices and vectors are in general defined over a mathematical *field*. The matrices and vectors we study in this week are defined over the field of real numbers. What this affects is our allowed domain for *scalars*. We accept real numbers as permissible scalar quantities, and imaginary or complex numbers as not. By extension, since the entries in matrices or vectors are by definition scalars, we allow only real-valued entries in them.

2.2 Definition of eigenvalue and eigenvector

For a square matrix A, $det(A - \lambda I)$ is called the ***characteristic polynomial*** of A. An matrix of dimensions $n \times n$ will in general have an n^{th} degree characteristic polynomial. From the Fundamental Theorem of Algebra, we know that an n^{th} degree polynomial with real coefficients has exactly n roots, including repetition. Some of these roots may be imaginary or complex. Complex roots always occur in conjugate pairs i.e. if $a + ib$ is a root then $a - ib$ will also be a root.

λ_j is called an ***eigenvalue*** of A iff. it occurs as a root of the characteristic polynomial of A i.e. it is a solution to the equation $det(A - \lambda I) = 0$.

For a matrix A, $\vec{v}$ is an ***eigenvector*** iff. $Av = \lambda v$ where λ is is an eigenvalue and $\vec{v} \neq \vec{0}$.

Alternatively, $Av = \lambda v$ can be rewritten as $(A - \lambda I)v = \vec{0}$ i.e. $\vec{v} \in \mathbb{N}(A - \lambda I)$.

2.3 Procedure for finding eigenvectors

1. Find the eigenvalues.
2. Find the corresponding eigenvectors.

2.3.1 Finding eigenvalues

Depending on the exact numbers in the matrix, one or the other of the following two methods will be easier.

1. Set up and solve the **characteristic polynomial**:

$$\boxed{det(A - \lambda I) = 0}$$

2. Set up these equations and solve for λ_js:
   (only as many as you need e.g. only 2 equations for a 2×2 matrix)

Properties of eigenvalues:

(a) $\Sigma\lambda_j = tr(A)$ where $tr(A)$ is the *trace* of A viz. the sum of diagonal values

(b) $\prod \lambda_j = det(A)$

(c) $\Sigma\lambda_j^2 = tr(A^2) \because$ if λ_j is an eigval of A then λ_j^k is an eigval of A^k where $k \in \mathbb{N}/\{0\}$

(d) $\Sigma\frac{1}{\lambda_j} = tr(A^{-1})$ if A is invertible

Worked out example Find the eigenvalues of A = $\begin{bmatrix}1 & 2\\ 2 & 3\end{bmatrix}$

1. $\lambda_1 + \lambda_2 = tr(A) = 4$
2. $\lambda_1 \times \lambda_2 = det(A) = -1$
3. Solving, we get $\lambda_1 = 2 + \sqrt{5}$ and $\lambda_2 = 2 - \sqrt{5}$

Remarks

- 0 appears as an eigenvalue iff. the matrix has a non-trivial null-space.
- Eigenvalues, the solutions to the characteristic polynomial, in general may be real, imaginary or complex.
  - But: For a special kind of matrix viz. real symmetric matrices, the eigenvalues are always real.
- **While we are limiting our definition of vectors and matrices to be over the field of** $\mathbb{R}$, when a matrix has one or more imaginary / complex eigenvalues, we can get no eigenvectors from them, and so we immediately know that for a matrix $A \in \mathbb{M}_{n\times n}$ we cannot construct an eigenbasis of of $\mathbb{R}^n$ w.r.t. A as it does not have n linearly independent eigenvectors.
  - if a matrix has *only* imaginary / complex eigenvalues, we'll say that the matrix has no eigenvectors. An example of such a matrix is:

$$\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}$$

with the characteristic polynomial: $\lambda^2 + 1 = 0$.

2.3.2 Finding eigenvectors

1. Find the eigenvalues of A.
2. (a) Write $(A - \lambda_j I)$ for each distinct λ_j.
   (b) Compute the row-echelon form $ref(A - \lambda_j I)$ using Gaussian elimination.
   (c) Use $ref(A - \lambda_j I) = \vec{0}$ to find the basis vector(s) for $\mathbb{N}(A - \lambda_j I)$. These basis vectors are the eigenvectors.

Worked out example :

$$\text{Find the eigenvectors of } \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}.$$

1. The eigenvalues are $\lambda_1 = 4$ and $\lambda_2 = 2$.
2. (a) $(A - \lambda_1 I) = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$

   (b) $ref(A - \lambda_1 I) = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}$

   (c) $\begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \implies \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

So the first eigenpair (λ_1, v_1) is $(4, \begin{bmatrix} 1 \\ 1 \end{bmatrix})$. I leave it to you as an exercise to find the second eigenvector following the steps above.

Remarks

- Eigenvectors for distinct eigenvalues are always linearly independent.
  - But: For a special kind of matrix viz. real symmetric matrices, eigenvectors for distinct eigenvalues are not only linearly independent, they are orthogonal. (by extension, the eigenspaces are orthogonal)
- But: if an eigenvalue is repeated, then there are ***at most*** as many linearly independent eigenvectors for that eigenvalue as the (algebraic) multiplicity of that eigenvalue. The span of these eigenvectors is called the *eigenspace* of that eigenvalue.
  - For an eigenvalue, the number of times it appears as a root in the characteristic polynomial is called its *algebraic multiplicity*.
  - For an eigenvalue, the number of linearly independent eigenvectors corresponding to it, equivalently the dimension of its eigenspace, is called its *geometric multiplicity*.
  - $\boxed{\text{geometric multiplicity} \leq \text{algebraic multiplicity}}$
  - For example the matrix $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ has the eigenvalue $\lambda_1 = 1$ with algebraic multiplicity of 2, but geometric multiplicity of 1 as there is only one linearly independent eigenvector $v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

2.4 Eigenvectors in action: approximating the n^{th} Fibonacci number

The Fibonacci sequence is given by:

$$\boxed{F_{k+2} = F_{k+1} + F_k}$$

Let:

$$\boxed{u_k = \begin{bmatrix} F_{k+1} \\ F_k \end{bmatrix} \text{ and } u_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}}$$

Then:

$$\boxed{u_{k+1} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_{k+1} \\ F_k \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}}_{\text{"}A\text{"}} u_k}$$

In general, the chain can be seen to be:

$$\boxed{u_o \xrightarrow{A} u_1 \xrightarrow{A} u_2 \xrightarrow{A} u_3 \dots}$$

and so:

$$\boxed{u_k = A^k u_0}$$

Can we avoid this expensive matrix exponentiation? **Yes**, if we can find an eigenbasis of $\mathbb{R}^2$ w.r.t. A. Can we? Yes.

A has the eigenvectors

$$\boxed{v_1 = \begin{bmatrix} \frac{1+\sqrt{5}}{2} \\ 1 \end{bmatrix} \text{ with eigenvalue } \lambda_1 = \frac{1+\sqrt{5}}{2}}$$

and

$$\boxed{v_2 = \begin{bmatrix} \frac{1-\sqrt{5}}{2} \\ 1 \end{bmatrix} \text{ with eigenvalue } \lambda_2 = \frac{1-\sqrt{5}}{2}}$$

These two eigenvectors are linearly independent, and can form a basis of $\mathbb{R}^2$, the input space of A.

So then, only to write u_0 as a linear combination of v_1 and v_2:

$$\boxed{\begin{aligned} &u_0 = (\tfrac{1}{\sqrt{5}})v_1 + (\tfrac{-1}{\sqrt{5}})v_2 \\ &= c_1 v_1 + c_2 v_2 \text{ (say)} \end{aligned}}$$

and the setup is complete.

$$\boxed{\begin{aligned} &u_k = A^k u_0 = A^k(c_1 v_1 + c_2 v_2) \\ &= c_1(A^k v_1) + c_2(A^k v_2) \\ &= c_1(A^{k-1}(Av_1)) + c_2(A^{k-1}(Av_2)) \\ &= c_1(A^{k-1}(\lambda_1 v_1)) + c_2(A^{k-1}(\lambda_2 v_2)) \\ &\vdots \\ &= c_1(\lambda_1^k v_1) + c_2(\lambda_2^k v_2) \end{aligned}}$$

Hence,

$$\boxed{F_k = \tfrac{1}{\sqrt{5}}(\tfrac{1+\sqrt{5}}{2})^k - \tfrac{1}{\sqrt{5}}(\tfrac{1-\sqrt{5}}{2})^k}$$

And as $(\frac{1-\sqrt{5}}{2})^k \to 0$ as $k \to \infty$:

$$\boxed{F_k \approx \tfrac{1}{\sqrt{5}}(\tfrac{1+\sqrt{5}}{2})^k}$$

3 Diagonalization

Motivation

Subtitle: *A matrix is diagonalizable, so what?*

1. If A is diagonalizable, then it is much easier (computationally inexpensive) to compute arbitarily higher powers A^k from the diagonalized form.
2. If A is diagonalizable as $A = SES^{-1}$ then A is similar (mathematically, by definition) to E. A "simple" form such as diagonal enables an instant lookup of the rank, eigenvalues, invertibility, is it a projection, etc. That is, all properties which are invariant under the similarity transform, are much easier to assess.
    - The matrix is invertible only if 0 does not appear as an entry in the diagonal matrix.
    - The diagonal entries in the diagonal matrix are the eigenvalues.

3.1 Definition of diagonalizability

A matrix A is diagonalizable if it can be factorized as $A = SES^{-1}$ where E is a diagonal matrix.

3.2 General conditions for diagonalizability

Necessary and Sufficient Condition (1)

A matrix $A \in \mathbb{M}_{n\times n}$ is **diagonalizable iff. it has n linearly independent eigenvectors**.

The above can be given an equivalent restatement as below:

Necessary and Sufficient Condition (2)

A matrix is **diagonalizable iff. for each eigenvalue its geometric multiplicity** (number of linearly independent eigenvectors associated with it, dimension of its eigenspace) **equals its algebraic multiplicity** (number of times it occurs as a root of the characteristic polynomial).

A matrix that does not meet the above conditions is not diagonalizable, and is called a *defective matrix*.

3.2.1 Special cases

Special case: Sufficient Condition when all distinct eigenvalues

For $A \in \mathbb{M}_{n\times n}$ if all n eigenvalues are distinct (and real, if we're working over the field of $\mathbb{R}$), then:

1. because distinct eigenvalues always have linearly independent eigenvectors, A has n linearly independent eigenvectors;
2. and so A is diagonalizable.

Special case: Real symmetric matrix

Any real symmetric matrix is diagonalizable.

3.3 Properties

(1) If a matrix A is diagonalizable as $A = SES^{-1}$ then:

the diagonal matrix E contains only the eigenvalues of A (with repetition if any), and S contains only the eigenvectors of A.

Hence, because the set of eigenvalues of A is unique, the diagonal matrix E of a diagonalizable matrix A is unique, *up to* the relative ordering of the eigenvalues. Note that the ordering of the eigenvectors in S (as column vectors, from left to right) is coupled with the ordering of the eigenvalues in E (down the diagonal, from top-left to bottom-right).

(2) If a matrix A is diagonalizable as $A = SES^{-1}$ then:

$$\boxed{A^k = SE^kS^{-1} \text{ for } k \in \mathbb{N}/\{0\}}$$

This is a **major** computational advantage gained with diagonalization. Recall that $[diag(a, b, c \ldots)]^k = diag(a^k, b^k, c^k \ldots)$ i.e. to exponentiate a diagonal matrix, you don't need to do the repeated multiplication, just exponentiate the diagonal elements. This allows computing arbitrary powers of A with a bounded computational complexity that doesn't scale with k nearly as sharply.

4 Orthogonal Diagonalizability

Motivation

Recall from our motivating eigenvectors, and then demonstrating with the Fibonacci sequence approximation, that applying a matrix A to vectors in its input space becomes much easier if we first express the vector in terms of the eigenbasis of the input space w.r.t. A.

Well, some bases are easier to work with than others. In particular, the easiest basis to work with is an *orthonormal basis*. Why? Just take the input vector you want to express, and compute dot products with the the orthonormal basis vectors, to get the respective coefficients. Only an orthonormal basis e.g. the standard normal basis like $\{(1, 0), (0, 1)\}$ affords this convenience.

So when a matrix is orthogonally diagonalizable, the column vectors in the left matrix do not just make up an eigenbasis, they make up an *orthonormal* eigenbasis, further simplifying dealings with the matrix.

4.1 Orthogonal matrices

$$\boxed{\begin{array}{l} Q \text{ is an orthogonal matrix if } Q^TQ = QQ^T = I \\ \implies Q^{-1} = Q^T \end{array}}$$

4.2 Orthogonal diagonalizability

$$\boxed{\begin{array}{l} A \text{ is } \textit{orthogonally diagonalizable} \text{ if } \exists Q \text{ such that Q is orthogonal and } A = Q\Lambda Q^T \text{ where } \Lambda \text{ is a} \\ \text{diagonal matrix.} \end{array}}$$

4.2.1 Special case: Real symmetric matrices

$$\boxed{\text{Any real symmetric matrix is orthogonally diagonalizable.}}$$

5 Real Symmetric Matrices

So far, we have brought up real symmetric matrices a number of times as having some special properties. Let us recapitulate them in one place:

1. Eigenvalues, the solutions to the characteristic polynomial, in general may be real, imaginary or complex. **But for a special kind of matrix viz. real symmetric matrices, the eigenvalues are always real.**

2. Eigenvectors for distinct eigenvalues are always linearly independent. But, f**or a special kind of matrix viz. real symmetric matrices, eigenvectors for distinct eigenvalues are not only linearly independent, they are orthogonal. (by extension, the eigenspaces are orthogonal)**

3. **Any real symmetric matrix is diagonalizable.**

4. More strongly, **any real symmetric matrix is *orthogonally* diagonalizable.**

In the subsequent weeks of this course, we will open up or expand the definition of vectors and matrices, from being over the field of $\mathbb{R}$, to being over the field of $\mathbb{C}$.

This leads to generalizations of the notions of "real symmetric matrices" and "orthogonal diagonalization" — viz. *Hermitian matrices*, and *unitary diagonalization*, respectively.

References

Wikipedia

Wikipedia contributors. *Vandermonde matrix (Wikipedia)*. URL: https://en.wikipedia.org/w/index.php?title=Vandermonde_matrix.
Wikipedia contributors. *Moore–Penrose inverse (Wikipedia)*. URL: https://en.wikipedia.org/w/index.php?title=Moore%E2%80%93Penrose_inverse.