Lecture Review

Ho Chi Vuong
AI Math Foundations: Abstract Vector Spaces
Center of Talent in AI

May 19, 2020

Lecture 1: Abstract Algebra & Vector Spaces

From Algebra to AI

This section starts off this course of with some brief history of AI, where its origin, algebra, traces way back in time, since the very start of civilization. We learn some famous figures in the development of AI, including Turin machine, Rosenblatt's PERCEPTRON and more.
This also touches on some intuition of some parts of machine learning, including basis functions and feature extraction (the analysis of the apple), manifold hypothesis, multimodal embeddings, encoder/decoder and the TEFPA (Task, Experience, Function space, Performance, Algorithm).

Abstract Algebra

In this part we review some key mathematics knowledge, including:

Coordinate systems

In a nutshell, coordinate systems exist to define a point in space. There are a lot of coordinate systems, but the most commonly used are Cartesian coordinates, Cylindrical coordinates, spherical coordinates and geographic coordinates.

Abstract Algebra

Algebra is the manipulation of symbols. Abstract are creating concepts that can be widely applied. Together, Abstract Algebra is the mathematics of creating concepts by the manipulation of symbols.

Set theory

Here we review on some properties of **Set**:

1. **Set** is a collection of objects
2. **Subset** is when a set is a part of a bigger set, then the smaller set is called a subset of the bigger set.
3. **Space** is a set with added structures. Here, vector spaces are sets with linear combination structures.

Affine space is a vector space where after translation retains the parallel structure of the original vector space (will be more on this on subsequent lecture).

Abstract Spaces

We are shown the diagram containing the dissection of the vector space $\mathbb{R}^n$ into different vector spaces that have different properties.

Vector Spaces & Linear Algebra

Linear combinations

$$x = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n$$

This is very useful in Linear Algebra, as it has some interesting properties such as decomposition (where we can understand what constitutes an object by each small objects that comprise it) and generation/synthesis (where we can create object by combining the desired properties).

Weighted sum and average

A good example of linear combinations is Weighted sum and average, wherein newer inputs are weighted higher than older inputs.

$$x = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n$$

with $x_1 < x_2 < \ldots < x_n$ as we input more x. This is useful in attention mechanism.

Vector spaces

Vector spaces are spaces with linear combination structures. By definition, vector space has the following properties:

1. **Vector addition** If $v_1, v_2 \in V$, then $v_1 + v_2 \in V$
2. **Scalar multiplication** If $v_1 \in V, c \in \mathbb{R}$, then $cv \in V$

Basis & Coordinate vector

Span of a set of vectors is a vector space that contain all possible linear combination of the vectors in the set. If $v = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$, then $\mathsf{span}(v) = \{w = w_1\mathbf{v}_1 + \ldots + w_n\mathbf{v}_n\}$.
We say that a set of vectors is **linearly dependent** when at least one vector in that set is in the span of the rest of the set, and **linearly independent** when no vector is in the span of the rest of the set. If $v = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$, then v is a linearly dependent set $\Leftrightarrow v_k \in \mathsf{span}(\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}, \mathbf{v}_{k+1} + \ldots + \mathbf{v}_n\}$
Basis is the smallest set of vector that can be used to represent a vector space by the span of that basis. If β is the basis of V, a vector space, then $V = span(\beta)$. Basis is linearly independent, and $\mathsf{dim}(\beta)$ is the dimension of that vector space.
Any vector belonging to $\mathsf{span}(\beta)$ can be written as $w = w_1\beta_1 + \ldots + w_n\beta_n$, with β being an **ordered basis**. We define coordinate vector as the vector representing w in terms of β. We have $[w]^\beta = \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}$.

Lecture 2: Linear Transformations

Transformations

The purpose of linear transformation is to reshape the input data into a more discernible, where each of the data can be easily formed into groups for generalizing concepts. In other words, transformation is *to disentangle the embedding manifolds*.

Linear & affine transformations

A transformation is linear when it retains its linear structure. If $T : V \to W$ is a linear transformation, then $T(au + bv) = aTu + bTv$, where $a, b \in \mathbb{R}$ and $u, v \in V$.
An **affine transformation** is simply a linear transformation that preserves its affine combination structure (for example, if two vectors in V are parallel, then they are still parallel after an affine transformation). Technically, it is just a linear transformation + translation.

Nullspace, range and rank of linear transformations

This section can be summed up with the Rank Nullity Theorem:

$$\mathsf{dim}(V) = \mathsf{dim}(\mathsf{range}(T)) + \mathsf{dim}(\mathsf{ker}(T))$$

with $T : V \to W$, $\mathsf{ker}(T) = \{v \in V : Tv = 0_V\}$, $\mathsf{range}(T) = \{v \in V : Tv \neq 0_V\}$

1. If $\mathsf{dim}(\mathsf{ker}(T)) = 0$, then T is injective (each $Tv = w$ is unique)
2. If $\mathsf{range}(V) = \mathsf{range}(W)$, then T is surjective (every $Tv = w$)
3. If both, then T is bijective (every $Tv = w$, with each $Tv = w$ is unique)

Matrix representation of linear transformations

Linear transformation can be expressed using matrices. If we have $T : V \to W$, then its equivalent in matrix form would be $A_T v = w$, with $A_T = ([T\beta_j]^\gamma)$, β being the basis of V, γ being the basis of W, $v \in V$ and $w \in W$.

Rank of a matrix & inverse matrices

In this we are acquainted with the concept of column and row space. For a matrix $A^{m \times n} = \mathbf{a}_1, \dots \mathbf{a}_n$ with $\mathbf{a}_1, \dots \mathbf{a}_n$ being column vectors, $\mathsf{Col}(A) = \mathsf{Row}(A^\mathsf{T}) = \mathsf{span}(\mathbf{a}_1, \dots \mathbf{a}_n)$.
We define $\mathsf{rank}(A)$ being the dimension of $\mathsf{range}(A)$. Then we have $\mathsf{rank}(A) <= \mathsf{min}(m, n)$. We say that a matrix is full rank when $\mathsf{rank}(A) = \mathsf{min}(m, n)$.
If $T : V \to W$ is a transformation, then we say $T^{-1} : W \to T$ is its inverse transformation when $T^{-1}T = I_V$ and $TT^{-1} = I_W$. To be invertible, that transformation must be **bijective**. An example of invertible matrix is Vandermonde matrix.

Projections

Details in lecture 3.

Lecture 3: Inner Product Spaces

Inner products & alignment/similarity

Inner products, dot products, norm and metric

To determine the similarity between two vectors, we can use **inner products**.
An **inner product** is defined as a transformation $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$. It has the following properties:

1. **Symmetry** $\langle u, v \rangle = \langle v, u \rangle$
2. **Linearity** $\langle \alpha u + \beta v, \gamma w + \omega z \rangle = \alpha\gamma\langle u, w \rangle + \alpha\omega\langle u, z \rangle + \beta\gamma\langle v, w \rangle + \beta\omega\langle v, z \rangle$
3. **Positive definiteness** $\langle v, v \rangle > 0$

A special form of inner product is **dot product**: $x \cdot y = \sum_{i=1}^{n} x_i y_i = \mathbf{x}^\mathsf{T}\mathbf{y} = \mathbf{y}^\mathsf{T}\mathbf{x}$. We also have the **generalized dot product** $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^\mathsf{T} M \mathbf{y}$, with M being a symmetric positive definite matrix (which means $\mathbf{x}^\mathsf{T} M \mathbf{x} > 0$)
The use of inner product is in convolutional operator, when a filter is scanned through the input data and we can have the map of domain, and in the case of picture, a **feature map**.

Inner product space

Inner product space is simply the vector space with inner product $\langle\cdot,\cdot\rangle$.
Normed vector space is the vector space with norm $\|\cdot\|$.
Metric space is the vector space with metric or distance.
There are many more structured vector space, like topological vector space, Hilbert and Banach spaces and so on.

Angle, length/size/norm, distance/metric

Unit vectors in direction of $\mathbf{a}$ is $\mathbf{u_a} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$.
Vector projection of $\mathbf{a}$ onto $\mathbf{b}$ is $\mathbf{a}_p = \|\mathbf{a}_p\|\mathbf{u}_b$. We then have the **scalar projection** $\|\mathbf{a}_p\| = \frac{\mathbf{a}^\mathsf{T}\mathbf{b}}{\|\mathbf{b}\|}$.
If $\mathbf{a}$ is **perpendicular** to $\mathbf{b}$, or $\mathbf{a} \perp \mathbf{b}$, then $\mathbf{a}^\mathsf{T}\mathbf{b} = \mathbf{b}^\mathsf{T}\mathbf{a} = 0$.
From the equation above, we can have $\cos(\theta) = \frac{\|\mathbf{v},\mathbf{w}\|}{\|v\|\|w\|}$, or the cosine similarity. We can then calculate the **angle** between $\mathbf{v}$ and $\mathbf{w}$.
Norm can be thought of as the *size* of a vector. The general equation of a $l_p - norm$ is $\mathsf{norm}_p(x) = \sqrt[p]{\sum_i |x_i|^p}$.
Metric is the distance between two vectors in a vector space. A *natural* metric is $d(u,v) = \|u - v\|$.

Orthogonality & projection

Orthonormal basis

An **orthonormal basis** is defined to be the basis of a vector space that is *mutuallyorthogonal* (all vectors are orthogonal to each other) and contained of **unit vectors** (or all vectors have its norm equals 1).
This makes it easy when wanting to find the orthogonal projection on that space, simply by **multiplying that vector with β, the orthonormal basis** of the concerned vector space.
Moreover, orthonormal basis also makes it easy to find **the coordinates of any vectors, inner products between any vectors, and the coordinate matrix of any linear transformation**.

Gramm-Schmidt orthogonalization

To acquire an orthonormal basis of a vector space, we can easily do the following. If $\beta = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ is a non-orthonormal basis of V, then we can change it to $\beta' = \{\mathbf{v}'_1, \mathbf{v}'_2, \dots, \mathbf{v}'_n\}$ by the following process:

1. Making every vector mutually orthogonal

$$\begin{aligned}
\mathbf{u}_1 &= \mathbf{v}_1 \\
\mathbf{u}_2 &= \mathbf{v}_2 - \mathsf{proj}_{\mathbf{u}_1}(v_2) \\
\mathbf{u}_3 &= \mathbf{v}_3 - \mathsf{proj}_{\mathbf{u}_1}(v_3) - \mathsf{proj}_{\mathbf{u}_2}(v_3) \\
&\vdots \\
\mathbf{u}_n &= \mathbf{v}_n - \sum_{j=1}^{n-1} \mathsf{proj}_{\mathbf{u}_j}(\mathbf{v}_k)
\end{aligned}$$

2. Unit-vector-ify each vectors

$$\begin{aligned}
\mathbf{v}'_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \\
\mathbf{v}'_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \\
\mathbf{v}'_3 &= \frac{\mathbf{u}_3}{\|\mathbf{u}_3\|} \\
&\vdots \\
\mathbf{v}'_n &= \frac{\mathbf{u}_n}{\|\mathbf{u}_n\|}
\end{aligned}$$

Orthogonal complements

We say that two subspace V and W is **orthogonal** to each other when $\forall \mathbf{v} \in V, \mathbf{w} \in W$, $\mathbf{v} \perp \mathbf{w}$.

4 fundamental subspaces of $A_{m\times n}$

$R(A) = C(A^\mathsf{T}) = span(\mathbf{a}_1, \dots, \mathbf{a}_n)$ when $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$. $C(A) = R(A^\mathsf{T})$ is the span of the columns of A. We can easily prove that $N(A) \perp R(A)$ and $N(A^\mathsf{T}) \perp C(A)$, and that $\mathbb{R}^n$ is the combined space of $N(A)$ and $R(A)$.

Lecture 4: Spectral Decompositions

Eigenvectors & Eigenspaces

Eigenvectors

We define **eigenvectors** as any vector $\mathbf{v} \in V$ in a transformation $T : V \to W$ that is $Tv = \lambda v$ with $\lambda \in \mathbb{R}$. In other words, the vector after transformation is dilated by λ.

Eigenspaces

An **eigenspace** is the set of vector $\{v \in V : Tv = \lambda v\}$, with each specific eigenvalue λ. Its nullspace is an eigenspace of T corresponding to $\lambda = 0$.

Diagonalizable matrix

A matrix A is **diagonalizable** when

1. A is a square matrix
2. There exists an invertible matrix P s.t. $P^{-1}AP$ is a diagonal matrix.

We then have $\Lambda = \mathsf{diag}(\lambda_1, \ldots, \lambda_n) = P^{-1}AP$. Symmetric matrix is always diagonalizable.
We can say that two matrices A and B are similar when $A = P^{-1}BP$

Determinants

A **determinant** of a matrix represents the change of volume on a object after the matrix transformation. The determinant of a full rank matrix is never 0.

SVD - Spectral Value Decomposition

SVD

Different from eigendecomposition (which is $\Lambda = P^{-1}AP$ and only applicable for square matrix), **Spectral Value Decomposition** can be used for any linear transformation. Namely,

$$A_{m\times n} = U_{m\times n}\Sigma_{m\times n}V^{\mathsf{T}}_{m\times n}$$

Intuitively, SVD has three steps: **rotate/flip, positive scale, rotate/flip again**.