More Linear Algebra

**Definition 1.1** (Dot Product).  *(Algebraic definition)* Let \( \mathbf{a} \) and \( \mathbf{b} \) be two vectors in \( \mathbb{R}^n \). Then the dot product (or inner product) between \( \mathbf{a} \) and \( \mathbf{b} \) is defined as:

\[
\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b} = \sum_{i=1}^{n} a_i b_i
\]

*(Geometric definition)* The dot product of two Euclidean vectors \( \mathbf{a} \) and \( \mathbf{b} \) is defined by

\[
\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos(\theta_{\mathbf{a}, \mathbf{b}})
\]

Also, The dot product \( \mathbf{w} \cdot \mathbf{x} = \mathbf{b} \) is a hyperplane, where \( \mathbf{w} \) is normal to it.

**Definition 1.2** (Projection). Let \( \mathbf{a} \) and \( \mathbf{b} \) be two vectors in \( \mathbb{R}^n \). The projection of \( \mathbf{b} \) onto \( \mathbf{a} \) is defined as:

\[
\text{proj}_a \mathbf{b} = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}|^2} \mathbf{a}
\]

The projection of a 2-D vector \( \mathbf{x} \) onto a 1-D line identified by unit vector \( \mathbf{u} \) is \( (\mathbf{x} \cdot \mathbf{u}) \mathbf{u} \). To project a \( N \)-D vector \( \mathbf{x} \) down to \( K \)-D \( \hat{\mathbf{x}} \), we have \( \hat{\mathbf{x}} = \sum_{i=1}^{K} (\mathbf{x} \cdot \mathbf{u}_i) \mathbf{u}_i \).

**Definition 1.3** (Outer Product). Let \( \mathbf{a} \) and \( \mathbf{b} \) be two vectors in \( \mathbb{R}^n \). Then the outer product (or tensor product) between \( \mathbf{a} \) and \( \mathbf{b} \) is defined such that

\[
\mathbf{a} \mathbf{b}^T = \begin{bmatrix}
    a_1 b_1 & a_1 b_2 & \cdots & a_1 b_n \\
    a_2 b_1 & a_2 b_2 & \cdots & a_2 b_n \\
    \vdots & \vdots & \ddots & \vdots \\
    a_n b_1 & a_n b_2 & \cdots & a_n b_n
\end{bmatrix}
\]

**Matrix Multiplication**

Let \( \mathbf{A} \in M_{n \times p}(\mathbb{R}) \) and \( \mathbf{B} \in M_{p \times m}(\mathbb{R}) \), then \( \mathbf{AB} \in M_{n \times m}(\mathbb{R}) \). Matrix multiplication \( \mathbf{AB} \) can be interpreted in two ways.

1) When we consider row vectors of \( \mathbf{A} \) and column vectors of \( \mathbf{B} \), the multiplication \( \mathbf{AB} \) can be viewed as

\[
\mathbf{AB} = \begin{bmatrix}
    \mathbf{Ab}_1 & \mathbf{Ab}_2 & \cdots & \mathbf{Ab}_m
\end{bmatrix}
\]

where \( \mathbf{B} = \begin{bmatrix}
    \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_m
\end{bmatrix} \). We know \( (\mathbf{Ab}_j)_i = \mathbf{a}_i^T \mathbf{b}_j \). Therefore, \( (\mathbf{AB})_{ij} = \mathbf{a}_i^T \mathbf{b}_j \).
2) When we consider column vectors of $A$ and row vectors of $B$, the multiplication $AB$ can be viewed as

$$AB = \sum_{i=1}^{p} a_i b_i^T$$

where $a_i b_i^T$ is the outer product with output dimension of $n \times m$. 

**Definition 1.4** (Orthogonal Matrix). An orthogonal matrix $Q$ is a square matrix with real entries whose columns and rows are orthogonal unit vectors (i.e., orthonormal vectors), i.e.

$$Q^T Q = QQ^T = I$$

Therefore, we have $Q^T = Q^{-1}$. To fully understand why Equation 7 holds, we need to know that for two orthogonal vectors $u_1$ and $u_2$, $u_1^T u_2 = 0$. And $u_1^T u_1 = |u_1|^2 = 1$. Therefore, in the resulting matrix, all entries are 0 except for ones along the diagonal.

**Probability**

Parts of this section reference [1].

**Event space**  We define a space $\Omega$ to be the set of all possible outcomes. An event space $S$ is the set of measurable events $\alpha$ such that $\alpha \in S$ and $\alpha \subseteq \Omega$ to which we are willing to assign probabilities. For example, if we roll a dice, then $\Omega = \{1, 2, 3, 4, 5, 6\}$. A possible event could be $\{1\}$ (we rolled one), $\{1, 3, 5\}$ (we rolled odd), etc. We say an event $\alpha$ happened if we observed an outcome $r \in \alpha$. The event space $S$ is closed under union ($\alpha \in S \land \beta \in S \rightarrow \alpha \cup \beta \in S$) and complementation ($\alpha \in S \rightarrow \Omega - \alpha \subseteq S$).

**Probability distribution**  Given $(\Omega, S)$, a probability distribution $\mathbb{P}: S \rightarrow \mathbb{R}$ is a mapping from events to real values, such that (1) for all $\alpha \in S$, $\mathbb{P}(\alpha) \geq 0$, (2) $\mathbb{P}(\Omega) = 1$, and (3) for $\beta \in S$, $\mathbb{P}(\alpha \cup \beta) = \mathbb{P}(\alpha) + \mathbb{P}(\beta) - \mathbb{P}(\alpha \cap \beta)$.

**Random variable**  A random variable $X: \Omega \rightarrow \mathbb{R}$ associates each outcome in $\Omega$ with a value. We use $\text{val}(X)$ to denote the set of possible values that $X$ can take. Random variables can be discrete or continuous. We primarily consider discrete ones. For simplicity, if $x, y$ are generic values for random variables $X$ and $Y$, then we write $\mathbb{P}(X = x, Y = y)$ as $\mathbb{P}(X, Y)$. For a specific value $x$, we write $\mathbb{P}(X = x)$ as $\mathbb{P}(x)$.

**Marginal distribution**  The marginal distribution over random variable $X$ is $\mathbb{P}(X)$.

**Joint distribution**  The joint distribution over random variables $X_1, \cdots X_n$ is $\mathbb{P}(X_1, \cdots, X_n)$ satisfying $\mathbb{P}(X_1) = \sum_{x_1, \cdots, x_n} \mathbb{P}(X_1, x_2, \cdots, x_n)$. Note that $1$ is arbitrarily chosen.

**Conditional probability**  For random variables $X, Y$, $\mathbb{P}(X, Y) = \mathbb{P}(X)\mathbb{P}(Y|X)$, where $\mathbb{P}(Y|X)$ is the probability of $Y$ conditioned on $X$. For random variables $X_1, \cdots, X_n$, we have $\mathbb{P}(X_1, \cdots, X_n) = \mathbb{P}(X_1)(X_2|X_1)\mathbb{P}(X_3|X_1, X_2)\cdots\mathbb{P}(X_n|X_1, \cdots, X_{n-1})$. 


Independence X and Y are independent if \( \mathbb{P}(X, Y) = \mathbb{P}(X)\mathbb{P}(Y) \). X and Y are conditionally independent given Z if \( \mathbb{P}(X|Y, Z) = \mathbb{P}(X|Z) \) or \( \mathbb{P}(X,Y|Z) = \mathbb{P}(X|Z)\mathbb{P}(Y|Z) \).

Bayes’s Theorem Because \( \mathbb{P}(X,Y) = \mathbb{P}(Y)\mathbb{P}(X|Y) = \mathbb{P}(X)\mathbb{P}(Y|X) \), we can write

\[
\mathbb{P}(X|Y) = \frac{\mathbb{P}(X)\mathbb{P}(Y|X)}{\mathbb{P}(Y)} \tag{8}
\]

where \( \mathbb{P}(X) \) is called the prior, and \( \mathbb{P}(X|Y) \) is called the posterior.

Expectation For discrete random variable \( X \), the expectation of \( X \) under distribution \( \mathbb{P} \) is defined as

\[
\mathbb{E}_\mathbb{P}[X] = \sum_x x\mathbb{P}(x) \tag{9}
\]

We can also write it as \( \mathbb{E}_{X} \). Note that we can show \( \mathbb{E}[f(X)] = \sum_x f(x)\mathbb{P}(x) \). When the subscript is not present, \( \mathbb{E}[f(X)] = \mathbb{E}_{X}[f(X)] \), and \( \mathbb{E}[f(X,Y)] = \mathbb{E}_{X,Y}[f(X,Y)] = \sum_x \sum_y f(x,y)\mathbb{P}(x,y) \), where \( X, Y \) means \( \mathbb{P}(X,Y) \), the joint probability.

Properties of expectation:

- \( \mathbb{E}_{\mathbb{P}}[aX + b] = a\mathbb{E}_{\mathbb{P}}[X] + b \)
- \( \mathbb{E}_{\mathbb{P}}[X + Y] = \mathbb{E}_{\mathbb{P}}[X] + \mathbb{E}_{\mathbb{P}}[Y] \)
- If \( X \) and \( Y \) are independent, \( \mathbb{E}_{\mathbb{P}}[XY] = \mathbb{E}_{\mathbb{P}}[X]\mathbb{E}_{\mathbb{P}}[Y] \)
- For a constant value \( c \), \( \mathbb{E}[c] = c \).

Conditional expectation We define \( \mathbb{E}_{\mathbb{P}}[X|y] = \sum_x x\mathbb{P}(x|y) \) as the conditional expectation (expectation of \( X \) given evidence \( Y = y \)).

Variance The variance of variable \( X \) is

\[
\text{Var}_{\mathbb{P}}[X] = \mathbb{E}_{\mathbb{P}}[(X - \mathbb{E}_{\mathbb{P}}[X])^2] = \mathbb{E}_{\mathbb{P}}[X^2] - (\mathbb{E}_{\mathbb{P}}[X])^2 \tag{10}
\]

Properties of variance:

- \( \text{Var}_{\mathbb{P}}[aX + b] = a^2\text{Var}_{\mathbb{P}}[X] \)
- If \( X \) and \( Y \) are independent, then \( \text{Var}_{\mathbb{P}}[X + Y] = \text{Var}_{\mathbb{P}}[X] + \text{Var}_{\mathbb{P}}[Y] \)

Covariance matrix Suppose \( X = [X_1, \ldots, X_n] \). Then we define the covariance matrix \( \Sigma \) of \( X \) as

\[
\Sigma = \mathbb{E}[(X - \mathbb{E}[X])(X - \mathbb{E}[X])^T] \tag{11}
\]

if \( \mu_i = \mathbb{E}[X_i] \), then each entry \( \Sigma_{ij} = \text{cov}(X_i, X_j) = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)] = \mathbb{E}[X_iX_j] - \mu_i\mu_j \).

\[\text{See http://www2.econ.osaka-u.ac.jp/~tanizaki/class/2012/econome1/05.pdf.}\]
Bayesian Optimal Classifier

Suppose \( D \) is some distribution of samples \((x, y)\), where \( x \in \mathbb{R}^d \) and \( y \in \text{val}(Y) \) and \( Y \) is a discrete random variable. This means \( D(x, y) \) outputs the probability of \((x, y)\) to exist in the world. A classifier \( f(x) \) outputs the category (or class) \( y \) given input \( x \). The **Bayesian optimal classifier** is one defined as

\[
    f^*(x) = \arg \max_y D(x, y)
\]

**Theorem 3.1.** Bayesian optimal classifier achieve minimal error among all classifiers.

**Proof.** Assume there exists another deterministic classifier \( f' \) that produces lower error than \( f^* \). Then, for some input \( x \), we have \( f'(x) \neq f(x) \). Suppose in the real world, input \( x \) can map to classes \( Y = \{y_1, \ldots, y_n\} \). Suppose \( f'(x) = y_p \in \mathcal{Y} \). We have:

- Probability of \( x \) to occur is \( D(x) = \sum_{y \in \mathcal{Y}} D(x, y) \).
- Probability of \((x, y_p)\) to be observed is \( D(x, y_p) = D(x, f'(x)) \).
- Probability of \((x, y_q)\) where \( y_q \in \mathcal{Y} \setminus \{y_p\} \) to be observed is \( D(x) - D(x, f'(x)) \). This is the probability that \( f' \) made a mistake.

Similarly, the probability that \( f^* \) made a mistake is \( D(x) - D(x, f^*(x)) \). By definition of Bayesian optimal classifier,

\[
    D(x, f^*(x)) = \max_y D(x, y) \geq D(x, f'(x))
\]

Therefore,

\[
    D(x) - D(x, f^*(x)) \leq D(x) - D(x, f'(x))
\]

Thus, \( f^* \) makes fewer mistakes. When \( f' \) and \( f^* \) only disagree on \( x \), it is not possible for \( f'(x) \) being correct while \( f^*(x) \) being wrong. Therefore, \( f^* \) is optimal. \( \square \)

**Perceptron**

Perceptron is one of the simplest (linear) binary classifiers. Suppose we observe data points \((x_i, y_i)\) for \( i = 1, \ldots, N \), where \( x_i \in \mathbb{R}^d \) and \( y_i \in \{-1, +1\} \). We hope to train the weights \( w \in \mathbb{R}^d \) and bias \( b \) in the following classification function:

\[
    \hat{y} = f(x) = \text{sign}(w \cdot x + b)
\]

To minimize a loss function \( L(w) = \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, \hat{y}) \). The perceptron algorithm is an iterative process, either offline or online. See Algorithm 1 and 2.

Perceptron is a linear classifier, which means the decision boundary is a linear combination of features (i.e. a hyperplane). Therefore, if the data is not linearly separable (cannot
Algorithm 1: Perceptron-Online(T)

1. $w^{(1)} \leftarrow 0; \quad b^{(1)} \leftarrow 0;$
2. foreach $t = 1, \cdots, T$ do
   3. $(x_t, y_t) \leftarrow$ new observation at time $t$;
   4. $\hat{y}_t \leftarrow f(w^{(t)} \cdot x_t)$;
   5. if $\hat{y}_t \neq y_t$ then
      6. $w^{(t+1)} \leftarrow w^{(t)} + y_t x_t$;
      7. $b^{(t+1)} \leftarrow b^{(t)} + y_t$
   8. end
9. end

Algorithm 2: Perceptron-Offline(D, T)

1. $w^{(1)} \leftarrow 0; \quad b^{(1)} \leftarrow 0;$
2. foreach $t = 1, \cdots, T$ do
   3. foreach $(x_i, y_i) \in D$ do
      4. $\hat{y}_i^{(t)} \leftarrow f(w^{(t)} \cdot x_i)$;
      5. if $\hat{y}_i^{(t)} \neq y_i$ then
         6. $w^{(t+1)} \leftarrow w^{(t)} + y_i x_i$;
         7. $b^{(t+1)} \leftarrow b^{(t)} + y_i$
      8. end
9. end
10. end

be separated by a hyperplane), then perceptron will not converge. If the data is indeed linearly separable, then perceptron is guaranteed to converge.

Prove that perceptron is guaranteed to converge.\footnote{Proof learned from Sham Kakade’s notes: \url{https://courses.cs.washington.edu/courses/cse546/16au/slides/notes09.pdf}. There is another proof [Novikoff] that shows a better upper bound for the number of mistakes, but it is not necessary for our problem.}

Assume the data is linearly separable. The definition of linear separability tells us that there exists some weights $w^*$ and the decision boundary $w^* \cdot x$ separates the data with a margin $\gamma \geq 0$, i.e.,

$$y_i (w^* \cdot x) \geq \gamma$$ \hspace{1cm} (16)

Suppose we have weights $w^{(t+1)}$ at time $t + 1$. We are interested to know if inequality $||w^{(t+1)} - w^*||^2 \leq ||w^{(t)} - w^*||^2$ holds. That is, $w^{(t+1)}$ is “closer” to $w^*$, the weights that can be used to separate the data. Let binary variable $m_i = 1$ if there is a mistake when
classifying data point $i$.

\[
||w^{(t+1)} - w^*||^2 = ||w^{(t)} + m_i y_i x_i - w^*||^2
\]

\[
= ||w^{(t)} - w^*||^2 + 2m_i y_i x_i^T (w^{(t)} - w^*) + m_i^2 y_i^2 ||x_i||^2
\]  

(18)

\[
\leq ||w^{(t)} - w^*||^2 + 2m_i y_i x_i^T (w^{(t)} - w^*) + m_i^2
\]  

(19)

\[
\leq ||w^{(t)} - w^*||^2 - 2m_i + m_i
\]  

(20)

\[
\leq ||w^{(t)} - w^*||^2 - m_i
\]  

(21)

(19) to (20) is because, from (16), we have $y_i x_i^T w^* \geq 0$. And because $m_i y_i x_i^T w^{(t)} \leq 0$, we have

\[
m_i y_i x_i^T (w^{(t)} - w^*) \leq m_i y_i x_i^T w^{(t)} - m_i \leq -m_i
\]  

(22)

Therefore,

\[
m_i \leq ||w^{(t)} - w^*||^2 - ||w^{(t+1)} - w^*||^2
\]  

(23)

Perceptron is indeed improving at every iteration. Suppose the total number of mistakes at iteration $T$ is $M_T = \sum_{i=1}^{T} m_i$. From the above inequality, we arrive at

\[
M_T \leq ||w^{(1)} - w^*||^2 - ||w^{(T+1)} - w^*||^2 \leq ||w^*||^2
\]  

(24)

The last inequality holds because $w^{(1)} = 0$. Therefore, there is an upper-bounded for the total number of mistakes, which means perceptron is guaranteed to converge.

**References**