CSE 4502/5717 Big Data Analytics
Fall 2019 Exam 3 Helpsheet

1. Let $f : \Re^n \to \Re$ be any function on n variables. Given a series of examples to learn f, we can fit them using a linear model: $f(x_1, x_2, \ldots, x_n) = w_1x_1 + w_2x_2 + \cdots + w_nx_n$. Linear regression computes the optimal values for the parameters by equating the gradient to zero. Let the examples be $(x_i^1, x_i^2, \ldots, x_i^n; y_i)$ for $1 \leq i \leq m$. Let $\boldsymbol{w} = (w_1\ w_2\ \cdots\ w_n)^T$ be the parameter vector. Also, let

$$\boldsymbol{X} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^n \\ x_2^1 & x_2^2 & \cdots & x_2^n \\ \cdots & & & \\ x_m^1 & x_m^2 & \cdots & x_m^n \end{bmatrix}.$$

   Then, we showed that the optimal value for $\boldsymbol{w}$ is $(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$ where $\boldsymbol{y} = (y_1\ y_2\ \cdots\ y_m)^T$.

2. An artificial neural network (ANN) can be thought of as a directed leveled graph $G(V, E)$ where each node corresponds to a neuron. If there is a directed edge from node i to node j, then, a signal flows from node i to node j.

   A simple example is a perceptron that has a single neuron. There are k inputs $x_1, x_2, \ldots, x_k$ to the neuron. Input x_i weighted by w_i, for $1 \leq i \leq k$. If the weighted input to the neuron, i.e., $\sum_{i=1}^{k} w_ix_i$, is $\geq \tau$ (τ being a threshold), then the output of the neuron will be 1; otherwise the output will be 0. We showed that any Boolean function can be represented with a NN consisting of multiple layers of perceptrons.

3. We considered forward and backward propagations in a general feed forward NN $G(V, E)$ and showed that both steps can be completed in $O(|V| + |E|)$ time (for each example). We also briefly discussed convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs). In addition, we discussed ways of improving the test accuracy. Specifically, we investigated ensemble learning, drop out method, and regularization.

4. **Association Rules Mining.** An **itemset** is a set of items. A **k-itemset** is an itemset of size k. A **transaction** is an itemset. A **rule** is represented as $X \to Y$ where $X \neq \emptyset, Y \neq \emptyset, X \cap Y = \emptyset$.

   We are given a database DB of transactions and the number of transactions in the database is n. Let I be the set of distinct items in the database and let $d = |I|$.

   For an itemset X, we define $\sigma(X)$ as the number of transactions in which X occurs, i.e. $\sigma(X) = |\{T \in DB | X \subseteq T\}|$ The **support** of any rule $X \to Y$ is $\frac{\sigma(X \cup Y)}{n}$. The **confidence** of any rule $X \to Y$ is $\frac{\sigma(X \cup Y)}{\sigma(X)}$.

   Association Rules Mining is defined as follows.

   Input: A DB of transactions and two numbers: minSupport and minConfidence.
   Output: All rules $X \to Y$ whose support is $\geq$ minSupport and whose confidence is $\geq$ minConfidence.

   An itemset is **frequent** if $\sigma(X) \geq n \cdot \text{minSupport}$

   We discussed the Apriori algorithm for finding all the frequent itemsets. This algorithm is based on the a priori principle: If X is not frequent then no superset of X is frequent. Also, If X is frequent then every subset of X is also frequent.

   The pseudocode for the Apriori algorithm is given next.

Algorithm 1: Apriori algorithm

```
k := 1;
Compute F_1 = {i ∈ I | σ(i) ≥ n · minSupport};
while F_k ≠ ∅ do
    k := k + 1;
    Generate candidates C_k from F_{k−1};
    for T ∈ DB do
        for C ∈ C_k do
            if C ⊆ T then
                σ(C) := σ(C) + 1;
    F_k := ∅;
    for C ∈ C_k do
        if σ(C) ≥ n · minSupport then
            F_k := F_k ∪ {C};
```

   We can use a hash tree to compute the support for each candidate itemset. We also discussed a randomized algorithm for finding frequent itemsets.

5. **Clustering:** We showed how to implement the hierarchical clusstering algorithm in $O(n^2)$ time on any input of n points. This algorithm repeatedly merges the two closest clusters until a target number of clusters is reached. To begin with each input point is a cluster on its own.