1 Biclustering (30 pts)

In this question you will analyze the complexity of graph problems related to biclustering.

a. Let $G = (V_1, V_2, V_1 \times V_2, w)$ be a weighted bipartite graph such that $w : V_1 \times V_2 \rightarrow R$. Prove that the problem of identifying the heaviest subgraph in $G$ is NP-complete.

b. Let $G$ be a bipartite graph where each VERTEX is assigned a weight (rather than edges as in the previous item). The maximum node biclique in $G$ is a complete subgraph such that the sum of weights of its vertices is maximum. Devise a polynomial algorithm for solving the maximum node biclique problem.

2 Precision-Recall and ROC Curves (30 pts)

Receiver Operator Characteristic (ROC) curves and Precision-Recall curves are commonly used to evaluate the performance in binary classification problems. ROC curve is a plot of the false positive rate (equation 1) on the x-axis and the true positive rate (equation 2) on the y-axis. Precision-recall curve is a plot of the true positive rate (also termed recall) on the x-axis, and precision (equation 3) on the y-axis.

\[
\text{FalsePositiveRate} = \frac{FP}{FP + TN} \quad (1)
\]

\[
\text{TruePositiveRate} = \frac{TP}{TP + FN} \quad (2)
\]
Precision = $\frac{TP}{TP + FP}$ (3)

where $TP$ is the amount of true positives, $FP$ is the amount of false positives, $TN$ is the amount of true negatives, and $FN$ is the amount of false negatives.

We say that one curve $\alpha$ outperforms another curve $\beta$ if $\alpha$ is always equal to or below $\beta$. Assume you have constructed two classifiers $a$ and $b$ for some classification problem with a fixed number of positive and negative examples. Prove that the ROC curve of $a$ outperforms the ROC curve of $b$ if and only if the Precision-Recall curve of $a$ outperforms the Precision-Recall curve of $b$.

3 Markov Clustering (40 pts)

Markov clustering is an algebraic method for detecting modules in graphs. The input is the connectivity matrix $M$ associated with the network ($M_{pq}$ is the weight of the edge from node $p$ to node $q$). In a preprocessing stage the connectivity matrix $M$ is normalized so as to have a stochastic form (i.e., the entries are positive $M_{pq} \geq 0$, and sum of each column is exactly one $M_{pq} \leftarrow M_{pq} / \sum_{i=1}^{n} (M_{iq})$).

The procedure for Markov clustering includes two consecutive steps:

Expansion: $M \leftarrow M^2$ and

Inflation: $M_{pq} \leftarrow M_{pq} / \sum_{i=1}^{n} (M_{iq})^r$ (where $r > 1$ is the inflation parameter). The expansion step simulates a random walk in the network while the inflation step increases the gap between strong and weak links (the rich get richer and vice versa). The algorithm starts with the normalized connectivity matrix and repeats the steps of expansion and inflation until convergence (i.e., until the operation of extension and inflation do not alter the matrix). The resulting matrix $M^*$, obtained upon convergence of the algorithm, is then used to extract the clusters. In this question we will investigate this matrix.

a. We say that a vector $v$ is homogenous if all its non-zero entries are equal. A matrix is called column homogenous if all of its columns are homogenous. Prove that $M^*$ must be column homogenous.

b. Let $G$ be the graph associated with the matrix $M^*$. For $s$, $t$, nodes in $G$, write $s \rightarrow t$ if there is an arc in $G$ from $s$ to $t$. Let $\alpha$, $\beta$, $\gamma$ be nodes in $G$. Prove the following statements:

1. $(\alpha \rightarrow \beta) \land (\beta \rightarrow \gamma) \Rightarrow (\alpha \rightarrow \gamma)$
2. \((\alpha \rightarrow \alpha) \land (\alpha \rightarrow \beta) \Rightarrow (\beta \rightarrow \alpha)\)

3. \((\alpha \rightarrow \beta) \Rightarrow (\beta \rightarrow \beta)\)

**Hint:** in this section it is enough to use the facts that \(M^*\) is idempotent under matrix multiplication (i.e., it is not altered by the expansion step) and stochastic.

c. A strongly connected component in a directed graph \(G\) is a maximal subgraph \(H\) such that for every ordered pair of nodes \(p, q\) in \(H\) there is a directed path (from \(p\) to \(q\)). Note that an isolated node \(p\) will be considered a strongly connected component only if it has a self edge (from \(p\) to itself).

A *star-like* component is a maximal subgraph \(H\) containing at least one strongly connected component \(C\) and all nodes \(p\) in \(G\) such that there is a path in \(G\) going from \(p\) to some element of \(C\). Use the relations in section \(b\) to prove that the graph associated with the matrix \(M^*\) is a collection of star-like components. These star-like components are exactly the clusters produced by MCL.