<https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20190000830.pdf>

**[0. Introduction]**

|  |
| --- |
| 이 논문은 **DTN에서의 routing problem을 machine learning을 이용한 분류 문제로 보고 해결**하는 것을 목적으로 한다.   * 이 방법은 **다양한 조건에서 적용**할 수 있다. (새로운 노드가 network에 들어오거나 기존 노드가 나가는 것 등) * 현재 **분명히 나타나지 않는 네트워크 트래픽 또는 disruption의 패턴**을 알 수 있다.   + Machine Learning을 통해, **네트워크 환경으로부터 얻은 데이터**를 이용하여 이러한 패턴을 발견할 수 있다. |

**[1. Machine Learning 알고리즘의 분류]**

|  |  |
| --- | --- |
| Supervised Learning | **학습을 위해 큰 dataset이 사용**된다.   * Learner는 새로운 instance를 분류하기 위해 **dataset에 기반한 rule을 만든다.** * Training set의 **데이터는 label**되어 있으며 learner는 **correct 또는 incorrect**에 대한 예측을 한다. |
| Unsupervised Learning | 알고리즘은 **데이터를 모델링하며 dataset의 association에 대해** 더 많이 알게 된다. |
| Reinforcement Learning | Learner는 각 상태에서 **결정을 하고, 그 결정에 대한 outcome의 값에 따라 reward를 받는** 방법으로 학습한다. (**trial-and-error method**)   * 이때 learner는 주어진 상태에서 **어떤 것이 good/bad인지에 대한 결정**을 내린다. |

**[2. 이 논문의 목표]**

|  |
| --- |
| 1. **delivery time을 최소화**한다.  2. **delivery probability를 최대화**한다.  “**time series prediction problem에 따른 destination**으로 **주어진 네트워크에서 가장 가능성이 높은 path를 학습**하는 문제를 해결한다.” |

* **Network condition, delay** 등에 의해 delay time이 달라질 수 있으므로 **Time을 DTN의 metric으로 하면 결정적인 결과가 나타나지 않는다**.
* **Delivery predictability는 routing의 성공을 측정할 수 있는 요소**이지만, 많은 경우에서 Source node에서는 **메시지가 deliver되었는지 알 수 없다**.

**[3. 입출력 데이터 설계]**

|  |  |  |
| --- | --- | --- |
| **입력값: Bundle이 어떤 경로를 따라 전송될 것인지를 결정하는 요소**   * 현재 및 미래의 **네트워크 구조**   + **Source, destination, relaying node**에서의 **neighboring node의 집합** * **Contact period의 duration** * **Data rate** * **Buffer capacity** * 각 neighbor의 위치   입력값 중 전체 또는 일부가 예측 기능한 주기성을 따를 것으로 기대할 수 있지만, **각 입력값은 시간에 따라 달라진다.**   |  | | --- | | **Attribute vector X에서의 입력값:**  **Time index** in the epoch, **Source** node, **Destination** node, **Message was delivered** or not (0 or 1) |   **Epoch:** **length of time을 epoch**로 하고 **각 epoch는 1일->24시간의 time slice**로 나누어진다.   * **각 node는 각 epoch에서 이동성 및 데이터 생성의 패턴**을 가질 것이며, 그 패턴은 반복될 것이다.   **출력값:**   |  | | --- | | **Attribute vector Y에서의 출력값:**  **Set of nodes** that **message was forwarded to**   * n개의 bit로 인코딩되어 있으며, **n은 네트워크에 있는 node의 개수**이다.   + 메시지가 해당 node를 방문한 경우 1, 아니면 0 | |

**[4. 학습 및 성능 측정 방법]**

|  |
| --- |
| **학습 방법:**   * Classifier는 **전송된 각 메시지에 대한 historical value**들과 **각 메시지에 대응되는 forward node string**을 이용하여 학습한다. * **Test data**의 subset은 **모델을 validate**하는 데 사용된다. * **X attribute의 string(time, source and destination)**만이 모델에 주어지고, 모델은 이것을 이용하여 **메시지가 forward될 가능성이 가장 높은 node의 set**에 대한 예측을 출력한다.   **성능 측정 방법:**   * **Test set의 actual output Y와 prediction output을 비교**하여 측정한다. |

* 이 모델은 **routing 분류 문제를 n개의 세부 문제로 나누고**, 이때 어떤 **node가 주어진 route에 따른 set of node의 member인지 나타내는 binary output**을 생성하는 하나의 classifier를 이용한다.

**[5. 학습 알고리즘 선택]**

|  |
| --- |
| **Naive Bayes, Decision Tree, K-Nearest Neighbors**의 3가지 classifier를 선택하여 어떤 것의 성능이 가장 좋은지 측정한다.   * **이 Classifier들의 attribute로 사용될, 자주 방문되는 node**를 알아내기 위해 **K-means clustering algorithm**을 사용한다. |

**[6. 구현]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[Training]**  각 node에 있는 데이터는 각 learning epoch에서 **여러 개의 time index (slices of time)으로 나누어진다**.   * Classifier의 sample은 **{Delivery indicator (0 or 1), Source node number, Destination node number, Forwarded node number and time index}**로 포맷된다. * Data는 다음과 같이 split된다.   **X = {Delivery status, Source, Destination, Time index}**   * **각 node에 대하여 output variable Y**가 생성되는데, 다음을 의미한다.  |  |  | | --- | --- | | 1 | Bundle은 해당 node로 forward되었다. | | 0 | otherwise |  * **숫자 데이터는 정규화되고 center되며 scale**되며,이때 **각 attribute는 대략적으로 표준정규분포화**된다.   + Python 라이브러리 Scikit-learn을 이용한다. * 데이터는 **K-fold cross validation**을 통해 **5개의 test and train set**으로 나누어진다.   + 그 후 **각 classifier는 각 node에 대해 한번 training data에 fitting**되고, K-fold를 통해 **각각 분리된 test set**을 이용하여 정확도로 점수화된다.   **[Separate Training Phase]**  Emulation period 동안, **Node의 위치에 따른 region을 할당**하기 위해 **K-means clustering**이 사용된다.  **[After training]**  모델은 **이후에 사용할 것을 대비하여 저장**되는데, 이때 Python Pickle persistence 또는 HDF5 파일을 이용하여 재학습하지 않는다.   * 학습된 모델은 IBR-DTN routing module에 있는 **제안된 path에 대한 새로운 예측**을 위해 runtime에서 import될 수 있다. |

**[7. 평가]**

|  |  |
| --- | --- |
| 평가 수식 (Metric) | 설명 |
| **Hamming Loss:** | 잘못 분류된 label의 비율   * **각 label을 기반으로 점수를 산출**하는, 더 관대한 lenient metric |
| **Zero-one loss:** | Prediction label이 **하나라도 잘못되면 전체 prediction이 잘못**되었다고 판단 |
| **Micro-average precision:** | Using **True Positive () and False Positive ()** |
| **Micro-average recall:** | Using **True Positive () and False Negative ()** |
| **Micro-averaged F1 score:** | Using weighted average of **Micro-average precision and recall** |