

# 현대차

## < Assignment >

[개발코드, 논문리서치] (국내/국외) medium, github 등

1. 주식분석 시뮬레이션 알고리즘(매수/매도 시점에 따른 최적화)
2. 주식분석에 적용가능한 시계열알고리즘 / 시계열 클러스터링(성능지표 포함)
3. 주식분석에 적용가능한 자연어 분석(SNS, 뉴스 연관성 등) 키워드 추출, 문서분류

## 0. 현대차의 주식 패턴

### 1) 현대차 3개월



5월 19일: 현대차 삼성배터리 이용 기사 보도(91,9 -> 99,1/ 거래량 폭등)

6월 12~15일: 펠리셰이드 생산 중단(협력사 사망사고, 11일:108,5 -> 15일: 97,0)

6월 24일: 노사공동선언(98,7 -> 103,0)

7월 15일: 그린뉴딜(정부; 수소차 기반 충전소 확대의지 표명/ 101,5 -> 109,0)

7월 17일: 정몽구회장 입원치료, 현대차 리콜, (16일: 110,0 -> 20일: 121,5)

7월 20일: 실적달성에 대한 기대감과 정부가 최근 발표한 '한국판 뉴딜'

7월 23일: 영업이익반토막(코로나) & 어닝서프라이즈 & 임금협상 종결(118,5 ->124,5)

8월 6일: 영국 모빌리티 기업과 UAM 인프라 조성 < 실적회복전망,판매호조 (134, -> 144,5)

8월 10일: 아이오닉공개(147, -> 170 / 11일: 179,)

8월 12일: 문제사항 발견 X, 하지만 연일하락(172, -> 158,) 기관투자자 매도세 예상

8월 19일: 자율주행 합작사 모셔널 (158, -> 164,5)

8월 24일: 미국 소비자 첨단 기술 만족도 조사(JD파워) 1위(156, -> 161,5)

8월 25일~28일: 뉴코나 라인업, '바이온', 제네시스 브랜드 강화(166, -> 172,)

8월 31일: 친환경 모빌리티 전략, BTS 아이오닉 음원 공개, 기관매수(172, -> 176,5)

## 2) 현대차 1년



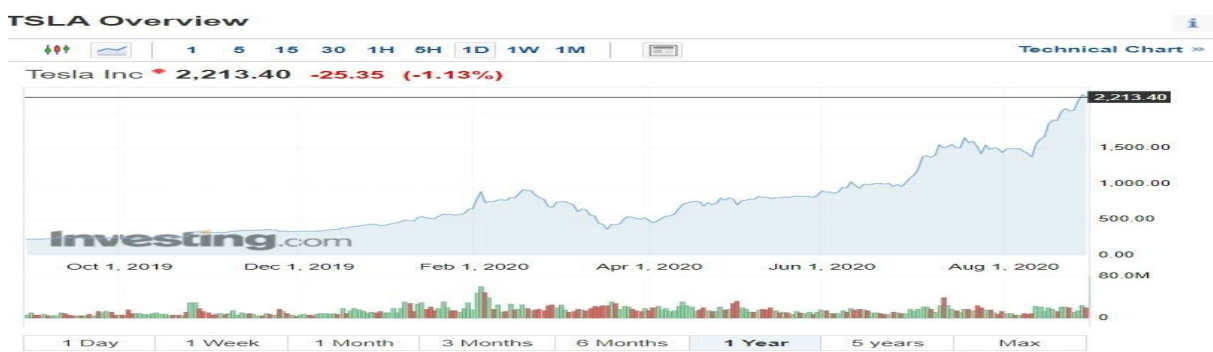
### ㉔ 차량 출시(단, 주가는 1000단위 표시)

GV80 : 1월 15일 출시 // 1월 22일 : 117, -> 127 ; 출시 전 주식 변동이 없음.  
 G80 : 3월 30일 출시 // 3월 25일 : 74,8 -> 84,5 ; 코로나 후 30일은 오히려 하락(86,4->85,4)  
 CN7 : 4월 7일 출시 // 4월 9일 : 90,0 -> 97,1 / 7일은 소폭 상승(88,8 -> 90,4)  
 산타페 HEV : 4/24 출시 무제한 연기 기사 // 4월 16일 : 99,1 / 21일 : 93,5 / 24일 : 90,1 ↓  
 아반떼 HEV : 8월 13일 // 8월 6일 : 144,5 / 10일 : 170,0 / 11일 : 179,0 / 13일 : 170,0 ↓  
 리콜 기사 : 2월 13일, 4월 15일, 5월 27일, 7월 2일 // 주가에 큰 영향 없음. ?

### ㉔ LG화학 1년 : 배터리 관련성

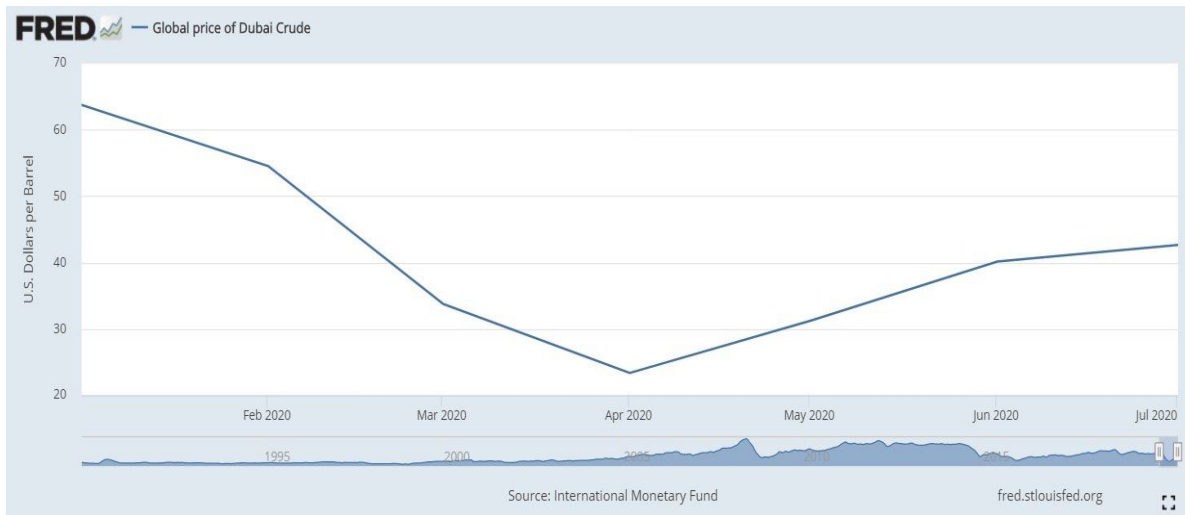


### ㉔ 테슬라 1년 : 경쟁업체(?) 진입

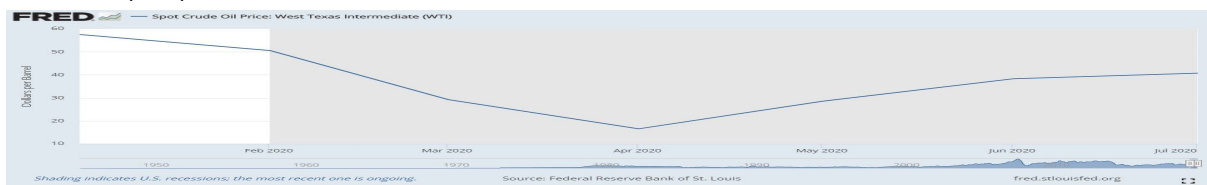


## ㉔ 유가변동

### < 두바이유(1년) >



### < 중질유(1년) >



## 3) 현대차 3년



## ㉔ 유가변동

### < 두바이유(3년) >



### < 중질유(3년) >



#### 4) 현대차 5년



#### ㉔ 환율(달러/원)변동



#### - 현대차 주가 움직임의 패턴 정리

- ① 정책에 영향받는 정도가 큼
  - 그린뉴딜에 수혜(수소차, 전기차). 이로인해 단순한 신제품 출시보다 정책 경향성에 맞는 제품 출시를 발표하는 경우 등락폭이 큼
- ② 신 제품 출시에 큰 영향(다만 단순한 리뉴얼 출시의 경우 주가 영향이 적음)
  - 예측을 위해 정책에 부합하는 신제품의 출시일이나 기업전략의 타이밍을 알아내야 할 것.
- ③ 제품 출시(주가상승)보다 제품의 문제(주가하락)가 발생시 주가에 미치는 탄력성이 크다
  - $(-)\Delta \text{stockP}(\text{신제품출시}) > (+)\Delta \text{stockP}(\text{펠리세이드 문제발생})$
- ④ 순간순간의 사건(리콜)보다 전체 기업 펀더멘탈이 주가에 전체적인 영향을 미침
  - 부분적 마찰이 많았으나 큰 틀에서 주가에 영향이 적은 경우가 많음
- ⑤ 단순한 기술 공동개발, 합작 계획이 주가에 미치는 영향은 크지 않다.
- ⑥ 유가변동, 환율이 주가와 비슷한 양상을 보인다.
  - 큰 경향성을 보일 수 있으나, 가중치 조정이 필요할 것

# 1. 주식분석 시뮬레이션 알고리즘

## (1) 은닉 마코프

### 1) 모형

- 순차적인 데이터 (sequence data)에 대해 현재 상태는 과거의 상태에만 영향을 받는다는 가정으로 현재 상태를 추정하는 확률모델
- 각 상태가 마코프 체인\*을 따르되 은닉 (hidden)되어 있다고 가정
- 숨겨져서 보이지 않는 은닉 상태값 (S)과 실제로 관측이 가능한 관측값 (O) 두 가지 요소로 이루어진 모델로 각 요소들 간의 움직임이 마르코프 체인을 따른다는 것
- 은닉 상태와 각 은닉 상태를 연결하는 전이 확률로 구성
- 은닉 마르코프 모델의 추정을 위해서 확률 평가, 최적 상태 추정, 모델 파라미터 추정의 문제가 발생

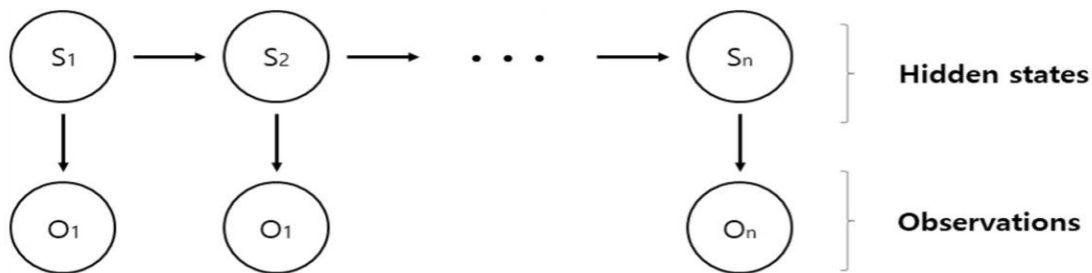


Figure 2.1 Hidden markov model trellis diagram

### 2) 모델 학습(R 프로그램 이용 : depmixS4 package)

- 주식시장에서 발생가능한 국면을 은닉상태로 정의
- 개별 종목의 주가 수익률을 은닉마코프 모델에 적용
- 입력받은 데이터를 기반으로 은닉 마코프 모델을 통해 각 은닉상태에 해당할 상태확률 (A)과 전이 확률 (B)을 산출
- 주식시장에서 상승, 횡보, 하락 국면이 존재함을 반영하여 은닉상태 (N)를 3개로 가정
- 관측값 (O)만 주어진 상태에서 은닉 마르코프 모델의 최적 파라미터를 결정
- 패키지 내의 EM(Expectation-Maximization) 알고리즘 중 하나인 Baum-Welch 알고리즘(학습 알고리즘)을 사용하여, 우도  $P(O|\lambda)$  값을 최대화하는 모델  $\lambda = (A, B, \pi)$ 의 각 파라미터를 학습
- 관측값 (O)는 주가 데이터이고, 각 모델의 파라미터는 관측값 (O)이 각 은닉상태에서 관측될 확률 (B)과 다른 은닉 상태로 전이될 확률 (A)을 추정
- 종목별로 slice duration은 1년으로 설정하여 데이터를 학습하고, slice delta를 1개월 설정하여 Window(이동간격)를 이동하며 학습

$\lambda$ : 은닉 마르코프 모델

$N$ : 은닉 상태의 수

$S = \{S_1, S_2, \dots, S_N\}$ : 상태의 집합

$A = a_{i,j}$ : 과거의  $i$ 상태에서 현재의  $j$ 상태로 전이 확률

$a_{i,j} = P(q_{t+1} = S_j | q_t = S_i)$ ,  $q_t$  시간  $t$ 에서의 상태

$B = b_i(O)$ :  $i$ 상태에서 관측값 ( $O$ )이 관측될 확률

$b_i = P(O | q_t = S_i)$

$\pi = \pi_i$ : 은닉 상태의 초기 확률

$\lambda \equiv (A, B, \pi, N)$

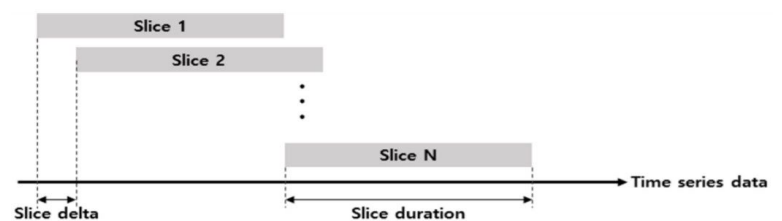


Figure 3.2 Sliding window method

### 3) 평가 및 포트폴리오 구성

- 각 상태별로 평균적인 sharp Ratio를 측정
- sharp ratio가 음수라면 리스크 대비 수익률이 음수이기 때문에 침체. 0에 가깝다면 횡보국면. 양수라면 상승 국면이라고 판단
- 개별 주식의 가격 모멘텀은 대표적으로 사용되는 직전 1개월 ~ 12개월의 주가상승률로 측정
- 기존 모멘텀 전략과의 수익률 비교 결과 장기적으로 은닉마르코프모델의 전략이 우수함을 알 수 있다.
- 기타 지표: Carhart 4 factor 모형(MKT[시장수익률], SMB[대형주 대비 소형주의 초과 수익률], HML[성장주 대비 가치주의 초과수익률], UMD[주가 모멘텀 팩터])

$$R_e = \alpha + \beta_1 MKT + \beta_2 HML + \beta_3 SMB + \beta_4 UMD + \epsilon.$$

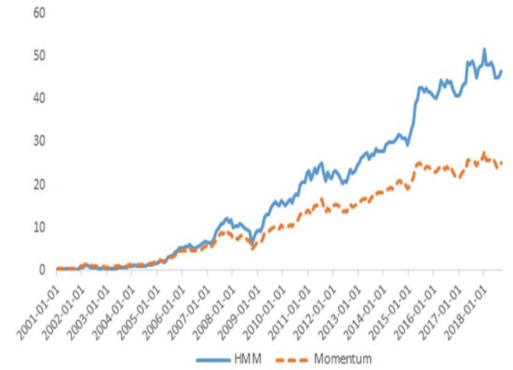


Table 4.2 Carhart 4 factor model regression result

	HMM	Momentum
Alpha	0.441** $t = 1.84$	0.421** $t = 1.999$
MKT	0.939*** $t = 25.998$	1.0444*** $t = 31.114$
HML	0.232*** $t = 3.944$	0.2699*** $t = 5.209$
SMB	0.368*** $t = 7.499$	0.3632*** $t = 8.839$
UMD	0.167*** $t = 3.407$	0.06*** $t = 1.393$
Observations	213	213
R <sup>2</sup>	0.77	0.82
Adjusted R <sup>2</sup>	0.76	0.85
Residual std. error	3.187	2.811
F-Statistic	172.837	248.850

Note : \* $p < 0.1$ ; \*\* $p < 0.05$ ; \*\*\* $p < 0.01$

## (2) 유전자 알고리즘

### 1) 모형

- 진화과정과 유전법칙을 모방하여 확률적 최적값 도출
- 세대마다 모집단 개체들의 Fitness평가
- 초기화 -> 적합함수 정의-> 선택 -> 교차 -> 돌연변이 -> 반복 및 종료
- ① 적합도함수 : 각 염색체들의 문제해결 적합도 평가
  - => ㉓ max 샤프 비율, ㉔ min 분산
  - (단, 샤프비율 : (포트폴리오수익률- 무위험수익률) / 표준편차)
- ② 염색체들의 적합도 평가, 낮은 염색체 제외
- ③ 선택, 교차, 돌연변이 연산 -> 유전자 조작 -> 최적 적합도 개체 찾기
- ④ 유전자 알고리즘의 파라미터는 인구 수, 세대수, 교배율, 변이율 설정
- ⑤ 최적화할 대상은 포트폴리오 종목의 비중

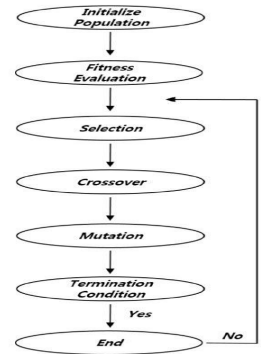


Figure 2.1 Genetic algorithm process

### 2) 함수 설정

- 목적함수는 sharp ratio의 최대화로써 유전자 알고리즘을 이용하여 최적화
- 제약조건 : 포트폴리오 투자 비중의 합이 100%가 되도록 함
- 각 종목의 투자비중에 비음수 제약조건을 주어 공매도를 제한
- 비중의 상한선을 0.5로 두어 전체 포트폴리오의 50% 비중을 넘지 않도록 함

### 3) 매매시점

- 종가데이터 기준, '입력변수X가중치'를 합산, 일자별 스코어 계산, 평균스코어 이용, 평균스코어가 거래기준스코어보다 클 경우 '매수'

### 4) 평가

- 기존 마코프 전략에 비해 수익률 상승

$N$ : 포트폴리오 구성 종목의 수  
 $w$ : 포트폴리오 비중 벡터 ( $N \times 1$ )  
 $\Pi$ : 포트폴리오 공분산 매트릭스 ( $N \times N$ )  
 $\mu$ : 구성종목별 기대수익률 벡터 ( $N \times 1$ )  
 $u$ : 투자 상한비율  
 $r_f$ : 무위험자산의 수익률  
 $J$ : 성분이 모두 1인 벡터 ( $N \times 1$ )  
 Object function = Maximize  $\frac{\mu^T w - r_f}{\sqrt{w^T \Pi w}}$   
 subject to  $w^T J = 1$   
 $u \geq w \geq 0$ .

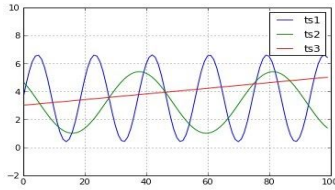
Table 4.3 HMM and HMM with GA portfolio summary

	HMM with GA	HMM	KOSPI200
Ann ret (Arith)	0.326	0.251	0.107
Ann ret (CAGR)	0.327	0.242	0.809
Ann std dev	0.292	0.256	0.204
Ann sharp	1.119	0.944	0.438
Win ratio	0.593	0.584	0.570
Maximum draw down	0.428	0.463	0.485



## 2. 시계열 클러스터링

### (1) 기존 유클리디언의 한계



In the above example, it is clear that  $ts1$  and  $ts2$  are most similar (they are both  $\sin$  functions under different transformations).  $ts3$  is clearly the most different. Let's compute the Euclidean distance  $d(ts1, ts2)$  and  $d(ts1, ts3)$  to see if the Euclidean distance measure agrees with what our intuition tells us. Let's first create a function that computes the Euclidean distance between two time series.

```
In [35]: def euclid_dist(t1, t2):
         return sqrt(sum((t1-t2)**2))
```

Let's now find the Euclidean distance between  $ts1$  and  $ts2$

```
In [36]: print euclid_dist(ts1, ts2)
26.959216038
```

and the Euclidean distance between  $ts1$  and  $ts3$

```
In [37]: print euclid_dist(ts1, ts3)
23.1892491903
```

This is not good because according to the Euclidean distance measure,  $ts1$  is more similar to  $ts3$  than to  $ts2$  which contradicts our intuition. This is the problem with using the Euclidean distance measure. It often produced pessimistic similarity measures when it encounters distortion in the time axis. The way to deal with this is to use dynamic time warping.

#### - 문제

- ① 두 time의 길이가 다른 경우
- ② 두 time series의 모양이 얼추 비슷하다고 판단하여야 할 경우

### (2) 보완적 접근

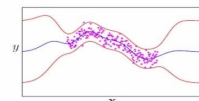
#### 1) 가우시안

- 가우시안 프로세스: 확률변수의 평균과 커널함수의 분포
- $X \rightarrow Y$ 인  $f$ 의 distribution을 구하게 된다. 이 때  $N$ 개의  $(X, Y)$  data set을 가지고 있다면,  $p(f)$ 는 multivariate Gaussian distribution을 갖는다고 가정.
- $p(f)$ 가 MVG를 따른다고 가정하였기 때문에  $p(f)$ 는 mean과 covariance로 나타낼 수 있음.
- distribution of function이기 때문에 각각을 mean function과 covariance function이라고 명명
- GP는 이러한 mean function과 covariance function을 찾는 작업.
- covariance를 구하는 것이 어렵기 때문에 covariance function이 특정한 형태(parameter)를 갖고 있을 것이라고 가정하고, 이 형태의 함수를 **kernel  $K(x, x')$** 이라고 부른다.

#### Nonlinear regression

Consider the problem of **nonlinear regression**:

You want to learn a function  $f$  with error bars from data  $\mathcal{D} = \{X, y\}$



A **Gaussian process** defines a distribution over functions  $p(f)$  which can be used for Bayesian regression:

$$p(f|\mathcal{D}) = \frac{p(f)p(\mathcal{D}|f)}{p(\mathcal{D})}$$

$$k(x_i, x_j) = \sigma_f^2 \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right) + \sigma_n^2 \delta(x_i, x_j)$$

위 식으로 계산된 커널 행렬을  $K$ 라고 할 때, 커널 함수의 하이퍼파라미터(hyperparameter)  $\theta = \{l, \sigma_f, \sigma_n\}$ 는 다음과 같은 로그 우도(log likelihood)가 최대가 되도록 추정된다.

$$\log p(y|X, \theta) = -\frac{1}{2}y^T K^{-1}y - \frac{1}{2}\log |K| - \frac{n}{2}\log 2\pi$$

## 2) K-means

- 커널 함수 및 추정된 하이퍼파라미터를 통해 평균, 분산 추정
- 평균 벡터에 대한 분류 모델
- 대표적인 **분리형 군집화 알고리즘**.
- 각 군집은 하나의 **중심(centroid)**을 가지며, 각 개체는 가장 가까운 중심에 할당되며, 같은 중심에 할당된 개체들이 모여 하나의 군집을 형성.
- 사용자가 사전에 군집 수(k)를 정해야 알고리즘을 실행할 수 있음.
- k가 **하이퍼파라미터(hyperparameter)**

$$X = C_1 \cup C_2 \dots \cup C_K, \quad C_i \cap C_j = \emptyset$$

$$\operatorname{argmin}_C \sum_{i=1}^K \sum_{x_j \in C_i} \|x_j - c_i\|^2$$

## 3) 분할-병합 기법

- K-평균 클러스터링 알고리즘이 지정된 k값에 대해 최적화된 결과를 제공하기 때문에, k개의 클러스터 가운데 다시 분할되거나 병합되는 경우 결과가 개선되는 클러스터들이 존재할 것을 가정
- 분할 : 클러스터들을 서브클러스터로 분할하는 과정(베이지 요인 검정 기준 충족), 클러스터 개수가 수렴할 때까지 수행
- 병합 : 분할 과정이 완료된 뒤 클러스터들 가운데 가장 평균 거리가 작은 클러스터 쌍에 대해 베이지 요인 검정을 수행. 클러스터 개수가 수렴할 때까지 반복

## 4) Dynamic Time Warping

- 속도가 다른 2개의 값이 들어올 때, 유사성을 측정하는 알고리즘
- DTW에서는 거리가 가장 짧은 시점(유사도가 높은 시작점)을 찾는다.

### Dynamic Time Warping

Dynamic time warping finds the optimal non-linear alignment between two time series. The Euclidean distances between alignments are then much less susceptible to pessimistic similarity measurements due to distortion in the time axis. There is a price to pay for this, however, because dynamic time warping is quadratic in the length of the time series used.

Dynamic time warping works in the following way. Consider two time series  $Q$  and  $C$  of the same length  $n$  where

$$Q = q_1, q_2, \dots, q_n$$

and

$$C = c_1, c_2, \dots, c_n$$

The first thing we do is construct an  $n \times n$  matrix whose  $i, j^{th}$  element is the Euclidean distance between  $q_i$  and  $c_j$ . We want to find a path through this matrix that minimizes the cumulative distance. This path then determines the optimal alignment between the two time series. It should be noted that it is possible for one point in a time series to be mapped to multiple points in the other time series.

Let's call the path  $W$  where

$$W = w_1, w_2, \dots, w_K$$

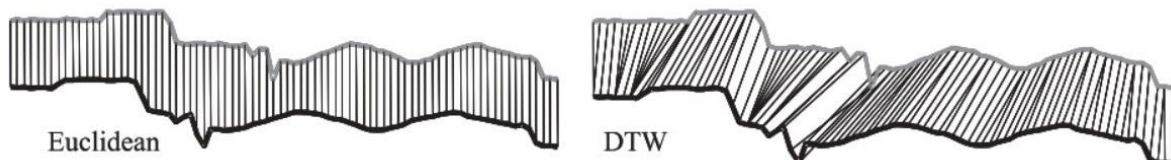
where each element of  $W$  represents the distance between a point  $i$  in  $Q$  and a point  $j$  in  $C$  i.e.  $w_k = (q_i - c_j)^2$

So we want to find the path with the minimum Euclidean distance

$$W^* = \operatorname{argmin}_W (\sqrt{\sum_{k=1}^K w_k})$$

The optimal path is found via dynamic programming, specifically the following recursive function.

$$\gamma(i, j) = d(q_i, c_j) + \min(\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1))$$



**Fig. 1.** Note that, while the two time series have an overall similar shape, they are not aligned in the time axis. Euclidean distance, which assumes the  $i^{th}$  point in one sequence is aligned with the  $i^{th}$  point in the other, will produce a pessimistic dissimilarity measure. The nonlinear dynamic time warped alignment allows a more intuitive distance measure to be calculated

**Input:** large scale news data gathered from web  
**Output:** 1) temporal trend charts of all keywords  
 2) similarity values between keywords  
 3) experimental result with knn classifier

**Functions :**

- func **MA()** #simple moving average algorithm
- func **ED()** #euclidean distance algorithm
- func **FastDTW()** #fast dynamic warping time algorithm

**Main:**

```

1 #step1 : building time series data according to generation options per keyword
2 build keyword_list from database storing web news data
3 for keyword in keyword_list:
4   fetch all frequency of keyword
5   make a baseline TS of keyword #TS = time series
6   calculate MA3_TS, MA5_TS, MA10_TS, MA20_TS by MA(TS) #window size
7   store all generated temporal data
8   visualize trend chart of each keyword
9
10 #step2: generating models applied by proposed method
11 for keyword in keyword_list:
12   making all pairs of keyword
13   if timeseries_type = baseline TimeSeries
14     calculate ED(TS, TS, of pairs) and FastDTW(TS, TS, of pairs)
15   else if timeseries_type in [MA3_TS, MA5_TS, MA10_TS, MA20_TS]:
16     for MTS in [MA3_TS, MA5_TS, MA10_TS, MA20_TS]:
17       calculate FastDTW(MTS, MTS, of pairs)
18   store sim_values_list of all pairs
19
20 #step3: evaluating the performance of pattern mining with knn classifier
21 #k-nearest neighbors algorithm
22 for keyword in keyword_list:
23   for k_param in [1, 3, 5, 7]: #four models of knn
24     get top k_param word sorted by sim_values_list
25     vote predicted_category of top k_param word
26     get final_category from max(predicted_category)
27     count results of final_category by comparing with origin_category
28     return comprehensive values of TP, TN, FN, FP
  
```

〈그림 4〉 전체 데이터 처리 과정 알고리즘

웹에서 수집된 대량의 뉴스 기사를 입력 받아(input), 1) 모든 키워드에 대한 추세 차트 그림, 2) 모든 키워드 쌍의 패턴 유사도 측정 결과, 3) 실험을 통한 모델별 분류 성능을 반환(output)하는 처리과정을 거친다. 전체 과정은 크게 3단계로 구분된다. 키워드별 시계열 데이터를 생성하는 단계, 제안 방법에 따라 다양한 분석 모델을 생성하는 단계, kNN 분류기를 통해 추세 패턴의 범주 별 분류 성능을 측정하는 단계이다.

프로세싱이 시작되면, 우선 웹으로부터 수집 저장한 뉴스 기사를 자연어 처리하여 키워드를 추출하고 리스트를 만든다(line 1). 모든 키워드의 빈도를 계산하여 120개 구간의 시계열 데이터를 생성한다(line 2-5). 기본 추세 데이터를 이용해 4가지 윈도우 크기가 적용된 이동 평균 데이터를 생성한다(line 6). 범주별 레이블링을 위해 추세 차트를 시각화한다(line 8).

두 번째 단계에서, 모든 키워드 데이터에 대해 조합쌍(combination)을 생성한다(line 11-12). ED와 DTW 함수를 이용하여 두 키워드 패턴 간의 유사도를 반복 계산한다. 이때 기본 시계열 데이터는 ED와 DTW 기법을 적용하고, 이동 평균을 적용한 4종의 데이터에 대해서는 DTW를 적용한다(line 13-17). 모든 과정이 끝난 후 다양한 모델별 유사도 측정 결과를 메모리(또는 데이터베이스)에 저장한다(line 18).

세 번째 단계에서, kNN 분류기의  $k(=1,3,5,7)$  값에 따라 성능 평가를 반복 수행한다(line 22-23). 두 번째 단계에서 얻은 두 키워드 간 유사도 측정 결과를 이용해 상위  $k$ 개 만큼의 레이블 데이터를 이용해 다수결 투표한다(line 24-26). 분류기가 예측한 결과와 레이블된 범주의 정답을 비교한다(line 27). 정확률, 재현율 평가를 위한 항목(TP, TN, FN, FP)을 반환하여 성능 평가를 실시한다(line 28).

## - What is the 'lb keogh'?

: LB\_Keogh is a tool for lower bounding various time series distance measures. It was introduced in 2002 as the first non trivial lower bound for Dynamic Time Warping (DTW), and it is still the fastest known technique for indexing DTW (see [9]). It can also be used to index under uniform scaling, various other distortions, and it can be used for efficient processing of streaming time series [8] and for indexing [shapes](#)

This function calculates a lower bound (LB) on the Dynamic Time Warp (DTW) distance between two time series. It uses a Sakoe-Chiba constrain

## - usage

```

: lb_keogh(x, y, window.size = NULL, norm = "L1", lower.env = NULL, upper.env =
  NULL, force.symmetry = FALSE, error.check = TRUE)
  
```

## 5) 나이브 베이즈

## 6) 서포트 벡터 머신

### 3. 자연어 분석, 키워드, 문서 분류

배터리

자동차 판매량

자동차 출시

리콜(7.17 : 110, -> 117,5 긴 장대 양봉?)

아이오닉(8.10 : 147, -> 170,)

개발협력(UAM 8.6 : 134, -> 144,5)

노사협정

플랫폼

그린뉴딜(7.14 : 100,0 -> 109,0)

전기

수소

삼성

e-GMP

<참고문헌>

『매매시점 파악을 위한 유전자 알고리즘 기반 스코어링 모델』

- 조기환, 정승환, 김경섭, 오경주/한국데이터정보과학회

『동적 시간 와핑 기법을 이용한 시계열 키워드 데이터 분류 성능 개선 방안』

- 정도현/정보관리학회

『은닉 마르코프 모델과 유전자 알고리즘을 이용한 자산배분전략 제안』

- 김은충, 오경주/한국데이터정보과학회

『투자자 정보, 기술적 분석 및 유전자 알고리즘에 기반한 국내 주식 포트폴리오 최적화에 관한 연구』

- 김병덕/KIF 연구보고서

『Lower-Bounding of Dynamic Time Warping Distances for Multivariate Time Series』

- Toni M. Rath and R. Manmatha/University of Massachusetts

『Deep learning-based feature engineering for stock price movement prediction』

- Wen Long, Zhichen Lu, Lingxiao Cui/ELSEVIER