



# [LAB-04] TruncatedSVD

---



## #01. TruncatedSVD 개요

---



### [1] 차원 축소 (Dimensionality Reduction)

원본 데이터 행렬을 직접 분해하여 저차원 잠재공간으로 표현하는 방법

- 크기가  $(n \times p)$  인 데이터 행렬  $X$
- $k$ 개의 성분으로 축소하면  $(n \times k)$  로 변환됨
- 단, 기존 변수를 제거하는 것이 아니라 **선형결합된 새로운 축으로** 변환하는 것



### (1) 선형대수 기반 차원 축소

방법	설명
PCA	공분산 행렬을 고유값 분해하여 분산이 최대인 축을 찾음
TruncatedSVD	원본 행렬을 직접 특이값 분해하여 상위 $k$ 개 성분만 유지



### (2) TruncatedSVD의 수학적 구조

데이터 행렬  $X$ 를 다음과 같이 분해함.

원본 데이터를 가장 잘 보존하는  $k$ 차원 구조를 추출하는 과정

$$X \approx U_k \Sigma_k V_k^T$$

기호	의미
$U_k$	관측치의 잠재 표현 ( $n \times k$ )
$\Sigma_k$	상위 $k$ 개 특이값
$V_k$	변수 로딩 행렬 ( $p \times k$ )



### [2] PCA와의 차이

구분	PCA	TruncatedSVD
평균 제거 필요	O	X
공분산 기반	O	X

희소행렬 처리	제한적	가능
계산 방식	SVD 활용	직접 SVD

TruncatedSVD는 공분산 행렬을 만들지 않고 원본 행렬을 직접 분해함.



### [3] TruncatedSVD의 특징

- 비지도 학습 기법으로 설명변수만을 대상으로 학습
- 데이터 구조 압축 목적
- 고차원·희소 데이터에 적용



### [4] 활용 예시

분야	활용 목적
텍스트 분석	LSA (잠재 의미 구조 추출)
추천 시스템	잠재 요인 기반 표현
카드 매출 분석	소비 구조 요인 발굴



## #02. 준비작업



### [1] 패키지 참조

```
from hossam import *
import numpy as np

from pandas import pivot_table, DataFrame, merge
from sklearn.decomposition import TruncatedSVD
from sklearn.preprocessing import StandardScaler
```



### [2] 데이터셋 가져오기

```
origin = load_data("card_useage")
origin.head()
```

카드 사용량에 대한 데이터 (출처: 서울빅데이터캠퍼스)

	기준일자		업종대분류		
0	2016-01-09	노원구	가정생활/서비스	20120	5
1	2016-01-09	서대문구	요식/유흥	324883	10
2	2016-01-31	은평구	요식/유흥	5149865	5
3	2016-02-06	서대문구	유통	237919	50
4	2016-02-08	서초구	요식/유흥	180074	5

## #03. 잠재요인 발굴하기



### [1] 구 × 업종 매출 행렬 생성

```
pivot_df = pivot_table(
    origin,
    index="가맹점주소(구)",
    columns="업종대분류",
    values="카드이용금액계",
    aggfunc="sum",
    fill_value=0      # NaN 값을 0으로 채움
)

pivot_df
```

업종대분류	가전/가구	가정생활/서비스	교육/학원	미용	스포츠/문화/레저	여행/교통	요식/유흥	유통	음/식료품	의료	의류/잡화	자동차
가맹점주소(구)												
강남구	662049	955700	0	0	91546	0	301850	8031854	0	169763	0	26156
강동구	0	0	125750	0	0	0	315884	0	0	0	0	0
강북구	236410	0	0	0	0	0	0	0	55330	0	0	0
강서구	0	30180	0	0	272626	0	197679	0	35210	0	0	0

관 악 구	0	0	0	0	482377	201200	158445	635792	0	0	0	0
광 진 구	0	0	0	0	0	0	57845	510042	0	0	0	0
구 로 구	61366	25150	0	16599	0	0	174038	635893	0	0	0	130780
구 리 시	0	0	0	0	0	0	0	0	0	0	0	0
금 천 구	0	0	0	0	0	75450	0	0	0	101012	0	0
노 원 구	0	20120	0	0	0	0	452197	0	0	57845	0	0
도 봉 구	0	0	0	0	0	0	67402	0	240434	0	0	0
동 대 문 구	0	0	0	0	0	0	0	0	12575	0	0	0
동 작 구	0	0	0	0	0	0	616074	124493	819085	62875	0	0
마 포 구	0	2885409	0	0	1218266	16096	1297790	1721769	255423	259045	0	47785
부 천 시	0	0	0	0	538210	0	121223	80480	0	0	0	0
서 대 문 구	128265	0	0	0	0	0	1575442	386103	0	411957	0	0
서 초 구	0	120720	0	21126	22635	0	893328	268703	0	1062839	0	0
성 동 구	0	0	0	0	0	150900	236762	527446	422520	0	0	10060
성 북 구	0	48037	0	1364498	948708	0	99594	150900	0	0	0	0
송 파 구	0	191140	0	0	31689	0	176553	0	99091	240937	0	0
	0	0	0	0	0	0	861066	50300	125750	251500	0	0

양천구												
영등포구	44767	0	40240	0	0	0	345551	504006	0	34204	0	1060324
용산구	0	1290648	0	284698	0	0	1172594	1113894	0	49797	181080	0
은평구	0	90540	0	0	0	0	5189099	12575	0	0	0	0
의정부시	0	0	0	0	0	0	440226	1873373	0	0	0	0
종로구	55330	0	0	40240	40240	0	18108	206783	0	407430	0	0
종구	700176	107139	0	0	0	0	3988237	576660	392340	848460	0	44264
중랑구	1312327	0	0	0	0	0	62875	90490	0	0	0	0



## [2] 로그 변환 (왜도 완화) + 데이터 스케일링

```
pivot_log = np.log1p(pivot_df)

scaler = StandardScaler()
scaled_data = scaler.fit_transform(pivot_log)
```



## [3] 전체 차원에 대한 SVD 분석 수행

```
# 최대 차원 설정
max_components = len(pivot_df.columns)

svd = TruncatedSVD(n_components=max_components, random_state=52)
svd.fit(scaled_data)

# 설명분산 비율
explained_var = svd.explained_variance_ratio_

# 누적 설명분산
cum_explained_var = np.cumsum(explained_var)

# 표 생성
```

```

variance_df = DataFrame({
    "Component": range(1, max_components + 1),
    "Explained_Variance_Ratio": explained_var,
    "Cumulative_Explained_Variance": cum_explained_var
})

variance_df

```

	Component	Explained_Variance_Ratio	Cumulative_Explained_Variance
0	1	0.239	0.239
1	2	0.132	0.371
2	3	0.123	0.495
3	4	0.098	0.592
4	5	0.080	0.673
5	6	0.075	0.747
6	7	0.066	0.813
7	8	0.054	0.867
8	9	0.045	0.913
9	10	0.033	0.946
10	11	0.021	0.967
11	12	0.018	0.985
12	13	0.012	0.997
13	14	0.003	1.000

| 분산 비율이 90%를 넘는 요인 수는 9개이다.

## [4] 분산 비율이 90%까지인 요인만으로 재학습

```

n_factors = 9
svd = TruncatedSVD(n_components=n_factors, random_state=52)
area_factors = svd.fit_transform(scaled_data)
area_factors[:5]

```

```

array([[ 3.21104902, -1.18045744,  0.66791854,  0.46115067, -0.08633886,
       -1.24345233,  0.67685339,  0.2452793 ,  1.2717643 ],
      [-1.69682133, -0.29089156,  1.82098385, -1.43885638,  2.43420759,
       1.97049816,  1.87234129,  0.39648396,  0.39809684],
      [-2.80289456, -0.59232514,  0.12958908, -0.95933817, -1.31785775,
       -1.48584988, -0.24735401,  1.10068454,  0.57593458],
      [-0.37628832,  0.35467836, -1.83346807,  0.01369646,  0.75356266,
       -0.87358575,  0.95509618,  0.74235879, -1.15110041],
      [-1.00328009,  0.31591403, -0.8753253 ,  3.08446537,  0.33558007,
       1.26528356, -0.13078302, -0.46858438,  0.94557008]])

```



## [5] 구별 잠재요인 점수

```
columns = []

for i in range(n_factors):
    columns.append(f"Factor_{i+1}")

columns
```

```
['Factor_1',
 'Factor_2',
 'Factor_3',
 'Factor_4',
 'Factor_5',
 'Factor_6',
 'Factor_7',
 'Factor_8',
 'Factor_9']
```

```
factor_df = DataFrame(
    area_factors,
    columns=columns,
    index=pivot_df.index
)

factor_df
```

	Factor_1	Factor_2	Factor_3	Factor_4	Factor_5	Factor_6	Factor_7	Factor_8	Factor_9
가맹점주소(구)									
강남구	3.211	-1.180	0.668	0.461	-0.086	-1.243	0.677	0.245	1.272
강동구	-1.697	-0.291	1.821	-1.439	2.434	1.970	1.872	0.396	0.398
강북구	-2.803	-0.592	0.130	-0.959	-1.318	-1.486	-0.247	1.101	0.576
강서구	-0.376	0.355	-1.833	0.014	0.754	-0.874	0.955	0.742	-1.151
관악구	-1.003	0.316	-0.875	3.084	0.336	1.265	-0.131	-0.469	0.946
광진구	-0.923	0.543	0.761	0.099	1.169	0.144	-0.944	-0.606	1.165
구로구	0.734	0.249	2.279	1.326	-0.607	-0.901	-0.747	1.994	-0.470
구리시	-2.465	0.413	-0.432	-0.799	-1.426	0.093	0.787	0.342	0.306
금천구	-1.747	-0.504	-1.718	-0.140	-2.431	1.562	0.854	-0.310	1.112
노원구	-0.818	0.691	-0.077	-0.693	0.611	-0.367	0.249	-0.129	-0.353
도봉구	-2.281	-0.057	-0.637	-0.661	0.743	-0.425	-0.773	0.398	-0.942
동대문구	-3.108	0.132	-0.743	-0.867	-0.587	-0.483	-0.127	0.934	0.133
동작구	0.521	-0.881	-0.976	-1.312	0.544	0.037	-0.949	-1.370	-0.357
마포구	3.148	-1.738	-2.448	1.151	0.828	0.520	0.087	0.813	0.665
부천시	-1.110	0.975	0.148	1.618	0.867	-0.339	0.320	-0.606	0.026

서대문구	0.577	-0.672	1.115	-0.256	-1.218	-0.363	-0.212	-1.978	-0.403
서초구	2.437	1.567	-0.422	0.056	0.270	-0.345	1.042	-0.719	0.183
성동구	-0.164	-2.046	-1.338	1.533	-0.602	1.756	-1.581	0.674	-0.769
성북구	1.035	2.576	-0.141	1.627	-0.344	-0.514	1.215	0.591	-1.066
송파구	1.353	-0.223	-2.348	-1.646	1.068	-0.832	1.131	-0.067	0.310
양천구	-0.163	-0.642	-0.670	-0.699	-0.054	0.028	-0.828	-1.433	-1.427
영등포구	0.974	-2.740	2.999	-0.244	-0.514	1.698	1.508	0.271	-1.112
용산구	3.626	3.776	0.266	-2.261	-0.937	2.175	-1.554	0.823	0.263
은평구	-0.654	0.978	0.559	0.508	1.341	-0.262	-0.720	0.240	-0.483
의정부시	-1.313	0.633	0.999	0.845	0.691	0.190	-1.085	-0.763	-0.261
종로구	0.967	0.895	0.780	0.707	-1.916	-0.849	1.087	-1.199	-0.358
종구	2.699	-2.358	0.321	-0.991	0.053	-1.190	-0.997	0.480	0.013
중랑구	-0.657	-0.175	1.814	-0.063	0.333	-0.968	-0.888	-0.397	1.783



## [6] 업종별 요인 로딩 확인 (해석 핵심)

```
loading_df = DataFrame(
    svd.components_.T,
    columns=[f"Factor_{i+1}" for i in range(n_factors)],
    index=pivot_df.columns
)

loading_df
```

	Factor_1	Factor_2	Factor_3	Factor_4	Factor_5	Factor_6	Factor_7	Factor_8	Factor_9
업종대분류									
가전/가구	0.137	-0.285	0.444	-0.015	-0.341	-0.430	-0.018	0.054	0.240
가정생활/서비스	0.394	0.168	-0.158	-0.040	0.184	-0.215	0.084	0.473	0.001
교육/학원	-0.036	-0.218	0.382	-0.176	0.256	0.487	0.510	0.123	-0.139
미용	0.246	0.486	0.129	0.092	-0.289	-0.009	0.107	0.194	-0.232
스포츠/문화/레저	0.208	0.142	-0.295	0.441	0.137	-0.203	0.500	-0.026	0.086
여행/교통	-0.009	-0.210	-0.364	0.431	-0.180	0.508	-0.098	0.080	0.302
요식/유흥	0.343	-0.004	0.140	0.134	0.528	0.062	-0.202	-0.262	-0.371
유통	0.349	-0.025	0.253	0.402	-0.014	0.084	-0.376	-0.276	0.030
음/식료품	0.001	-0.347	-0.451	-0.229	0.108	-0.179	-0.307	0.184	-0.351
의료	0.367	-0.158	-0.116	-0.286	-0.235	0.003	0.166	-0.453	-0.011
의류/잡화	0.208	0.393	0.030	-0.318	-0.161	0.400	-0.324	0.209	0.080
자동차	0.269	-0.461	0.180	0.186	-0.078	0.069	-0.045	0.511	-0.095
전자상거래	0.358	-0.162	-0.241	-0.172	-0.366	0.155	0.226	-0.160	-0.226

<b>주유</b>	0.323	-0.051	-0.064	-0.315	0.371	0.000	-0.015	-0.009	0.665
-----------	-------	--------	--------	--------	-------	-------	--------	--------	-------

해석예시: Factor\_1에서 가정생활/서비스, 전자상거래 로딩이 높음

## 인사이트: Factor별 상위 로딩 업종 기준 해석

요인 해석은 “각 Factor에서 절대값이 큰 로딩 업종”을 중심으로 해석해야 함.

( $\pm$  부호는 방향, 크기는 영향력 의미)

- 요인 해석은 절대적 정답 없음
  - 부호는 상대 방향 의미

Factor	높은 로딩 업종 (절대값 기준)	유형 요약	Structure_Strength 해석
F1	가정생활/서비스, 의료, 전자상거래, 유통, 요식/유흥, 주유	생활밀착 종합 소비형	전반적으로 중간~높은 강도. 일상 소비 업종들이 구조적으로 안정적 위치 차지
F2	미용(+), 의류/잡화(+), 자동차(-), 음/식료품(-)	패션·미용 중심 소비축	미용·의류는 비교적 높은 강도. 트렌드·개인소비 성향이 뚜렷
F3	가전/가구(+), 교육/학원(+), 음/식료품(-)	내구재·교육 중심형	교육/학원(0.907) 강도 매우 높음 → 구조적 영향력 큼
F4	스포츠/문화/레저, 여행/교통, 유통	여가·이동 중심형	여행/교통(0.872) 구조 강함 → 이동·활동성 소비 명확
F5	요식/유흥(+), 주유(+), 전자상거래(-)	외식·오프라인 소비형	주유(0.889), 요식/유흥(0.828) 강도 높음 → 오프라인 대면 소비 축 분명
F6	여행/교통(+), 교육/학원(+), 의류/잡화(+)	이동·교육 활동형	교육·여행 모두 강도 높음 → 활동 기반 소비 구조
F7	교육/학원, 스포츠/문화/레저	문화·청소년 활동형	스포츠/문화/레저(0.814) 안정적 구조 형성
F8	자동차(+), 가정생활/서비스(+), 의료(-)	자동차·생활서비스 특화형	자동차(0.797) 강도 높음 → 특정 기능 소비 중심
F9	주유(+), 여행/교통(+), 요식/유흥(-), 음/식료품(-)	차량·교통 중심형	주유(0.889) 매우 강함 → 교통·차량 축 독립성 높음



## #04. 예측 모형에 적용하기



# [1] 기본 구조

구별 소비 구조를 설명변수로 추가하여 구의 소비 구조를 예측 모형에 반영

```
df_model_1 = merge(origin, loading_df, left_on="업종대분류", right_index=True)
df_model_1.head()
```

	기준일자	가맹점주	업종대	카드이용금액계	카드이용	Factor_1	Factor_2	Factor_3	Factor_4	Factor_5	Factor_6	Factor_7	Factor_8
--	------	------	-----	---------	------	----------	----------	----------	----------	----------	----------	----------	----------

		소 (구)	분 류		건 수 계								
0	2016-01-09	노 원 구	가 정 생 활/ 서 비 스	20120	5	0.394	0.168	-0.158	-0.040	0.184	-0.215	0.084	0.473
54	2017-01-01	마 포 구	가 정 생 활/ 서 비 스	21629	20	0.394	0.168	-0.158	-0.040	0.184	-0.215	0.084	0.473
87	2017-10-27	구 로 구	가 정 생 활/ 서 비 스	25150	5	0.394	0.168	-0.158	-0.040	0.184	-0.215	0.084	0.473
100	2018-02-02	송 파 구	가 정 생 활/ 서 비 스	40240	35	0.394	0.168	-0.158	-0.040	0.184	-0.215	0.084	0.473
109	2018-04-17	성 북 구	가 정 생 활/ 서 비 스	48037	5	0.394	0.168	-0.158	-0.040	0.184	-0.215	0.084	0.473



## [2] 구 X 업종 적합도(벡터 내적값) 도출

```

Z = factor_df.values
Lambda = loading_df.iloc[:, :n_factors].values

structure_matrix = Z @ Lambda.T

structure_df = DataFrame(
    structure_matrix,
    index=factor_df.index,
    columns=loading_df.index
)
df_model_2 = df_model_1.copy()

```

```

df_model_2["Structure_Fit"] = df_model_2.apply(
    lambda row: structure_df.loc[
        row["가맹점주소(구)"],
        row["업종대분류"]
    ],
    axis=1
)

df_model_2.head()

```

	기준일자	가 맹 점 주 소 (구)	업 종 대 분 류	카드이 용금액 계	카 드 이 용 건 수 계	Factor_1	Factor_2	Factor_3	Factor_4	Factor_5	Factor_6	Factor_7	Factor_8
0	2016-01-09	노 원 구	가 정 생 활/ 서 비 스	20120	5	0.394	0.168	-0.158	-0.040	0.184	-0.215	0.084	0.473
54	2017-01-01	마 포 구	가 정 생 활/ 서 비 스	21629	20	0.394	0.168	-0.158	-0.040	0.184	-0.215	0.084	0.473
87	2017-10-27	구 로 구	가 정 생 활/ 서 비 스	25150	5	0.394	0.168	-0.158	-0.040	0.184	-0.215	0.084	0.473
100	2018-02-02	송 파 구	가 정 생 활/ 서 비 스	40240	35	0.394	0.168	-0.158	-0.040	0.184	-0.215	0.084	0.473
109	2018-04-17	성 북 구	가 정 생 활/ 서 비 스	48037	5	0.394	0.168	-0.158	-0.040	0.184	-0.215	0.084	0.473

## Structure\_Fit 해석

Structure\_Fit는 구의 소비 성향 벡터와 업종 특성 벡터의 내적값이며,

값이 클수록 해당 구에서 해당 업종이 구조적으로 적합함을 의미함.

## #05. 구 단위 창업 추천 업종 도출



### [1] 구 x 업종 적합도 행렬

```
# -----  
# 1. 행렬 준비  
# -----  
Z = factor_df.values # (구 × K)  
Lambda = loading_df.iloc[:, :n_factors].values # (업종 × K)  
  
# -----  
# 2. 구조 적합도 행렬 계산  
# -----  
structure_matrix = Z @ Lambda.T  
  
# -----  
# 3. DataFrame 변환  
# -----  
structure_df = DataFrame(  
    structure_matrix,  
    index=factor_df.index,      # 구  
    columns=loading_df.index   # 업종  
)  
  
structure_df.head()
```

업종 대분류	가전/ 가구	가정생활/서비스	교육/ 학원	미용	스포츠/ 문화/레저	여행/ 교통	요식/ 유흥	유통	음/식료품	의료	의류/ 잡화	자동차	전자상거래	주유
가맹점주소(구)														
강남구	1.935	1.366	-0.113	0.204	1.187	-0.104	0.466	1.117	-0.390	1.160	-0.471	1.509	0.764	1.710
강동구	-0.913	-0.578	3.603	-0.990	-0.671	-0.675	0.261	-1.370	-1.123	-0.816	-0.052	-0.234	-1.066	0.939
강북구	1.147	-0.608	-0.684	-0.600	-1.109	-0.544	-2.309	-1.616	0.565	-0.914	-0.535	-0.119	-0.882	-0.692
강서구	-1.102	0.957	-0.262	0.148	1.158	-0.359	-0.003	-1.281	1.185	-0.330	-0.714	-0.265	0.193	-0.534

관악구	-1.117	-0.770	-0.464	-0.356	1.272	2.436	0.001	0.968	-0.991	-1.095	-0.618	-0.261	-0.825	-0.496
-----	--------	--------	--------	--------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------



## [2] 구별 상위 3개 업종 추천

```

recommend_df = (
    structure_df
    .apply(lambda x: x.sort_values(ascending=False).head(3).index.tolist(), axis=1)
)

recommend_df = recommend_df.to_frame(name="Top3_Industries")

recommend_df

```

	Top3_Industries
가맹점주소(구)	
강남구	[가전/가구, 주유, 자동차]
강동구	[교육/학원, 주유, 요식/유흥]
강북구	[가전/가구, 음/식료품, 자동차]
강서구	[음/식료품, 스포츠/문화/레저, 가정생활/서비스]
관악구	[여행/교통, 스포츠/문화/레저, 유통]
광진구	[주유, 유통, 요식/유흥]
구로구	[자동차, 가전/가구, 미용]
구리시	[여행/교통, 교육/학원, 의류/잡화]
금천구	[여행/교통, 전자상거래, 의료]
노원구	[교육/학원, 음/식료품, 요식/유흥]
도봉구	[음/식료품, 요식/유흥, 교육/학원]
동대문구	[음/식료품, 여행/교통, 의류/잡화]
동작구	[음/식료품, 의료, 요식/유흥]
마포구	[여행/교통, 자동차, 스포츠/문화/레저]
부천시	[스포츠/문화/레저, 요식/유흥, 유통]
서대문구	[의료, 가전/가구, 유통]
서초구	[스포츠/문화/레저, 미용, 가정생활/서비스]
성동구	[여행/교통, 자동차, 음/식료품]
성북구	[미용, 스포츠/문화/레저, 가정생활/서비스]
송파구	[주유, 가정생활/서비스, 음/식료품]
양천구	[음/식료품, 의료, 요식/유흥]
영등포구	[교육/학원, 자동차, 가전/가구]
용산구	[의류/잡화, 미용, 가정생활/서비스]

은평구	[요식/유흥, 유통, 가정생활/서비스]
의정부시	[유통, 요식/유흥, 의류/잡화]
종로구	[미용, 의료, 가전/가구]
종구	[자동차, 가전/가구, 음/식료품]
중랑구	[가전/가구, 주유, 유통]

```

recommend_expanded = DataFrame(
    recommend_df["Top3_Industries"].tolist(),
    index=recommend_df.index,
    columns=["Rank1", "Rank2", "Rank3"]
)

recommend_expanded

```

	Rank1	Rank2	Rank3
가맹점주소(구)			
강남구	가전/가구	주유	자동차
강동구	교육/학원	주유	요식/유흥
강북구	가전/가구	음/식료품	자동차
강서구	음/식료품	스포츠/문화/레저	가정생활/서비스
관악구	여행/교통	스포츠/문화/레저	유통
광진구	주유	유통	요식/유흥
구로구	자동차	가전/가구	미용
구리시	여행/교통	교육/학원	의류/잡화
금천구	여행/교통	전자상거래	의료
노원구	교육/학원	음/식료품	요식/유흥
도봉구	음/식료품	요식/유흥	교육/학원
동대문구	음/식료품	여행/교통	의류/잡화
동작구	음/식료품	의료	요식/유흥
마포구	여행/교통	자동차	스포츠/문화/레저
부천시	스포츠/문화/레저	요식/유흥	유통
서대문구	의료	가전/가구	유통
서초구	스포츠/문화/레저	미용	가정생활/서비스
성동구	여행/교통	자동차	음/식료품
성북구	미용	스포츠/문화/레저	가정생활/서비스
송파구	주유	가정생활/서비스	음/식료품
양천구	음/식료품	의료	요식/유흥
영등포구	교육/학원	자동차	가전/가구
용산구	의류/잡화	미용	가정생활/서비스

은평구	요식/유흥	유통	가정생활/서비스
의정부시	유통	요식/유흥	의류/잡화
종로구	미용	의료	가전/가구
중구	자동차	가전/가구	음/식료품
중랑구	가전/가구	주유	유통