Day 1

Representation

- Bag of words models vs. distributed representation
- BOW 모델은 해석력이 있습니다.
- Word2Vec과 같은 distributed representation은 단어의 semantic similarity를 표현할 수 있습니다.

Document distance

- Cosine similarity가 Euclidean보다 더 적합합니다.

Logistic regression

- 각 클래스의 대표 벡터를 parameter θ에 학습합니다.
- Softmax regression은 클래스의 개수가 3개 이상인 logistic regression입니다.

Preprocessing framework

- Noise canceling → Tokenization(including pos) → Filtering → Term weighting

TF-IDF

- TF-IDF는 term weighting 방법 중 하나입니다. 언제나 만능은 아닙니다.
- TF-IDF의 목적은 정보력이 적은 단어의 영향력을 줄이기 위함입니다.

KoNLPy

- 형태소 분석 vs. 품사판별
- Out of vocabulary problem

n-gram

Day 2

한국어 어절구조

- (L, R)
- 명사+ [조사]/ 어근+ 어미/ …

Tokenizer의 필요성

- 분석하는 대상이 단어인가, 문서인가에 따라 정확히 단어를 잘라내야 할 수도 아닐 수도 있습니다.

통계 기반 단어 추출

- Cohesion / Branching entropy / Accessor variety / Word piece model

통계 기반 명사 추출

- 명사 오른쪽에 등장하는 subwords(R) 분포를 이용할 수 있습니다.

Day 3

사전 기반 품사 판별

- 품사 판별기/ 형태소 분석기는 다음의 모듈들로 구성됩니다.
  (1) 후보생성(generate candidates)
  (2) 후보평가(evaluate candidates)
  (3) 후처리

용언 추출

- 새로운 용언은 새로운 어미에 의하여 만들어집니다. (대화체)

Point Mutual Information (PMI)

- 상관성을 측정하는 대표적인 metric입니다.
- Infrequent patterns에는 민감합니다.

상대적 출현 비율

- PMI와 비슷합니다.
- 키워드 추출에 이용할 수 있습니다.

LASSO regression

- L1 regularization
- Lasso regression으로도 키워드를 추출할 수 있습니다.

Day 4

From logistic to feed forward network

- Linear inseparable한 데이터를 판별할 수 있습니다.

SVM

- 문서 판별에서는 Linear kernel이 RBF kernel보다 좋습니다.
- Document classification은 "LINEAR"입니다.

k-NN classifier

- 비슷한 reference data를 찾은 뒤, 이를 이용하여 판별을 수행합니다

Naïve Bayes

- 확률 기반 판별기로, logistic과 함께 대표적인 baseline으로 이용됩니다.

Decision tree

- Sparse vector에는 적합하지 않습니다

CRF

- Sequential labeling용 알고리즘입니다.
- Softmax regression의 확장입니다.

CRF+ 한국어띄어쓰기

- 학습데이터에 어느 정도 노이즈는 있어도 됩니다.

Day 5

Word2Vec

- Softmax regression으로 설명할 수 있습니다.

Doc2Vec

- Word2Vec에 document id를 가상의 단어로 추가합니다.

GloVe

- Word2Vec과 GloVe가 서로 보존하려는 정보의 차이가 있습니다.

FastText (unsupervised)

- 단어를 subwords로 표현할 수 있습니다.
- Spelling error에 둔감한 word representation 방법입니다

FastText (supervised)

- Task specific word embedding

MDS

- MDS는 distance matrix를 보존합니다

PCA

- PCA는 "방향적 경향"을 보존하는 차원변환 방법입니다.

Kernel PCA

- Kernel PCA는 "분포적 경향"을 보존하는 차원변한 방법입니다.
- Kernel은 데이터 간의 proximity로 해석할 수 있습니다.

Locally Linear Embedding

- 원 공간(X)에서의 이웃 구조를 보존합니다.

ISOMAP

- k-NN graph에서의 최단 경로를 보존합니다.
- Swiss roll data를 기억하세요.

t-SNE

- 원 공간(X)에서의 이웃 간의 유사도를 확률 P로 표현합니다. 이를 보존하는 새로운 공간 Y를 찾습니다.
- t-SNE는 Barnes hut 알고리즘을 이용해야 합니다.
- t-SNE의 학습 결과에서 가까운 점은 실제로 유사합니다. 하지만 떨어진 점이 유사하지 않다고 보장할 수는 없습니다.

Vector visualization

- 시각화에 정답은 없습니다.
- 시각화의 목적은 원하는 정보를 잘 전달하는 것입니다.

Bokeh

- matplotlib은 대표적인 python plotting library입니다.
- 최근에는 더 좋은 라이브러리들이 많이 제공되고 있습니다.

Neural word embedding as implicit matrix factorization (paper review)

- Word2Vec은 word-context matrix에 PMI를 적용한 뒤, SVD로 차원 축소를 한 것과 같습니다.
- Semantic similarity가 보존되는 word representation의 핵심은 co-occurrence입니다.

Day 6

Latent Semantic Indexing (LSI)

- SVD를 이용하여 행렬 A를 $A=U\Sigma VT$로 분해합니다.
- U의 중요한 성분만 취하는 truncated SVD를 이용하면 topical space로 문서와 단어를 표현할 수 있습니다.

Probabilistic LSI (pLSI)

- pLSI는 k개의 토픽이 존재하고, 각 토픽에서 단어가 발생할 확률이 있다고 가정합니다.
- $p(w,d)=p(d)\Sigma_z p(w|z)\times p(z|d)$를 학습합니다.

Latent Dirichlet Allocation (LDA)

- pLSI의 overfitting 문제와 학습 방법을 개선한 토픽모델링입니다.

pyLDAvis

- LDA 모델의 학습 결과를 시각적으로 표현합니다.
- Topic labeling의 기능과 PCA를 이용한 각 topic의 2차원 시각화 기능을 제공합니다.

Sparse Coding (SC)

- 딥러닝 이전의 representation learning을 위해 자주 이용되던 방법입니다.
- 계산 비용이 비쌉니다. 좋은 성능을 보여주지만, python에서는 빠른 구현체가 없습니다.

Nonnegative Matrix Factorization (NMF)

- LSI의 각 축이 반드시 독립(orthogonal)이라는 가정이 비현실적입니다.
- NMF는 각 축의 독립 가정을 완화하여 단어와 문서를 topical representation으로 표현합니다.

Named Entity Recognition (NER)

- 전통적으로 CRF가 가장 널리 이용된 문제입니다.
- 핵심은 named entity 앞/뒤에 등장하는 단어입니다.
- 학습용 데이터가 없을 때에는 Word2Vec을 이용하여 일부의 entity에 대하여 label을 부여할 수 있습니다.

Day 7

k-means

- 수렴이 빠릅니다. max_iter 설정을 작게 하세요.
- Uniform effect가 발생합니다. k는 기대값보다 크게 설정하세요. 후처리로 중복된 군집을 묶는 것이 현실적인 방법입니다.
- k-means++라는 initializer는 저차원에서 잘 작동합니다.
- 고차원 sparse data에서 효율적인 initializer를 다뤘습니다.
- Sillouette은 고차원 데이터에 적합하지 않습니다.

Spherical k-means

- distance metric을 cosine으로 이용하는 k-means clustering입니다.
- 문서 군집화에는 가장 현실적인 solution입니다.

Gaussian Mixture Model

- k-means와 대부분 비슷하지만, covariance 모양을 조절할 수 있습니다.

Bayesian GMM

- k를 데이터 기반으로 찾아주지만, 노이즈가 많다면 잘 작동하지 않습니다.
- "모델의 fitness measure"와 "사람이 좋다고 생각하는 기준"의 gap

Hierarchical clustering

- 복잡한 모양 데이터의 군집화를 위한 알고리즘입니다
- Outliers에 둔감한 장점이 있습니다.

DBSCAN

- 밀도(density) 기반 군집화 방법입니다.
- Threshold에 민감합니다.
- 고차원에서는 밀도가 잘 정의되지 않습니다.
- "고차원에서는 가깝다는 말 외에는 의미가 없습니다"

Clustering labeling

- 키워드 추출 방법을 이용하면 k-means 군집화 결과로부터 cluster labels 을 추출할 수 있습니다.

Random Projection

- Randomly generated mapper를 이용하여 원 공간(X)에서의 거리를 보존하는 저차원 벡터(Y)를 만들 수 있습니다.

Locality Sensitive Hashing (LSH)

- Random projection을 이용하여 k-nearest neighbors를 빠르게 찾습니다.
- 대량의 데이터에서 k-NN을 찾을 때 유용합니다.

Inverted index

- Sparse data에 대하여 cosine 기준으로 유사한 벡터를 찾을 때 유용합니다.

String distance

- Levenshtein (edit) /Cosine / Jaccard distance 등이 이용됩니다.
- 한글의 경우 초/중/종성을 분해하거나 unit을 다르게 정의할 수 있습니다.

Inverted index + Edit distance

- Edit distance 기준으로 가까운 단어를 찾기 위해 inverted index를 응용할 수 있습니다.

Day 8

Graph

- Vector space보다도 유연하게 데이터 간의 관계를 표현할 수 있습니다.

PageRank / HITS

- Graph에서 중요한 마디를 찾는 알고리즘입니다.
- bias를 조절하면 personalized PageRank를 만들 수 있습니다.

TextRank

- Word co-occurrence graph나 sentence similarity graph를 만들어 PageRank를 적용합니다.
- Keyword / key-sentence 추출이 가능합니다.

KR-WordRank

- Substring graph에서 중요한 마디는 keyword입니다.

SimRank

- 두 마디가 가까이 위치하는지를 측정하는 방법입니다.
- Small world phenomenon에 의하여 similarity matrix가 dense해집니다.
- Single vector SimRank가 이용 가능한 형태입니다.
- Doc-term bipartite graph에서 topically similar words를 찾을 수 있습니다.

Random Walk with Restart

- 한마디 a 주변에 위치하는 다른 마디들을 탐색합니다.
- Propagation illustration을 기억하세요.
- Doc-term bipartite graph나 term-co-occurrence graph에서 topically similar words를 찾을 수 있습니다.

Ford algorithm (shortest path)

- 그래프에서의 두 마디를 연결하는 경로 중 가장 짧은 경로를 탐색합니다.
- Dijkstra와 Ford algorithm이 대표적인 풀이법입니다. Edge weight가 음수가 될 수 있다면 반드시 Ford를 이용해야 합니다.
- Word segmentation이나 Part-of-Speech tagging을 shortest path 문제로 정의할 수 있습니다.

Day 9

PyTorch

- data / model / loss / optim
- nn.Sequential/ 각 layer 별로 함수를 구현/ forward 함수 구현

Convolutional Neural Network

- Convolution (filter, stride, padding)
- Pooling : max-pooling
- Activation function, activation map
- 각 필터는 각 관점으로 이미지를 해석하는 것

Word-level CNN

- Yoon Kim의 word-level CNN 구조
- CNN filters는 n-gram 역할

Character-level CNN

- 첫 convolution layer는 tokenizer의 역할
- 두 번째 convolution layer는 n-gram 역할

Day 10

Recurrent Neural Network

- Sequence 형태 데이터의 특징을 반영한 network입니다.
- Tagging / sentence classification에 이용될 수 있습니다.
- Sentence generation에도 이용될 수 있습니다.

Sequence to sequence

- 번역처럼 한 문장을 context로 압축한 뒤, 이를 이용하여 다른 문장을 생성합니다.

Attention models

- 한 문장을 하나의 벡터로 표현하는 것은 비효율적이기 때문에, 문맥을 선택적으로 이용할 수 있도록 attention 개념을 도입했습니다.

Self-attention models

- Feed-forward network로도 sequence를 모델링 할 수 있습니다.
- "Encoding meaning"