Active Learning Literature Survey

- Burr Settles -

Hyunsik Kim

Contents

1. What is Active Learning?
2. Scenarios -- Learner와 Oracle의 관계 프로세스를 정의하는 방법론
3. Query Strategy Frameworks -- 데이터를 쿼리하는 방법론
4. Related Research Areas
5. Additional Discussions

What is Active Learning?

모델(learner)이 학습 과정에서 특정 데이터에 대해서, 모른다!! 라고 이야기하고, 학습자(oracle) 이 이를 알려주는 일련의 과정을 통해, 상대적으로 적은 training datasets 양에 대해, 좋은 성능을 내는 방법론

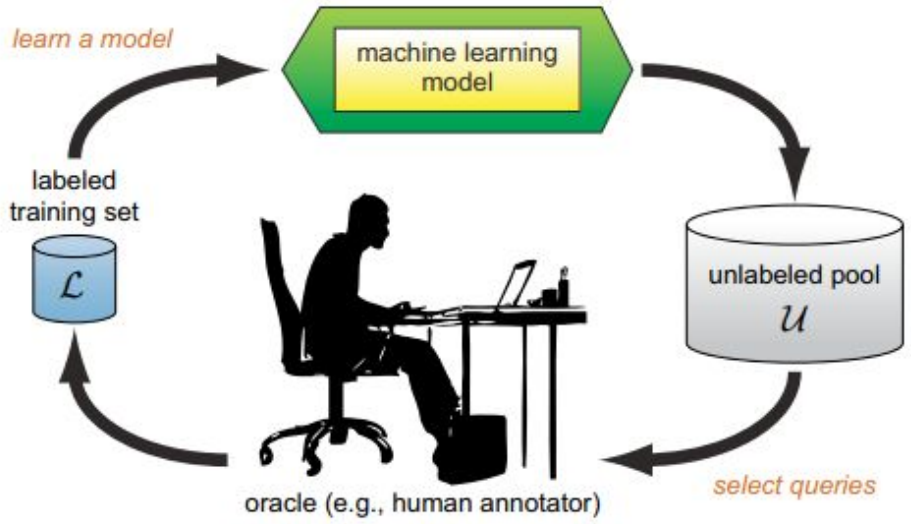
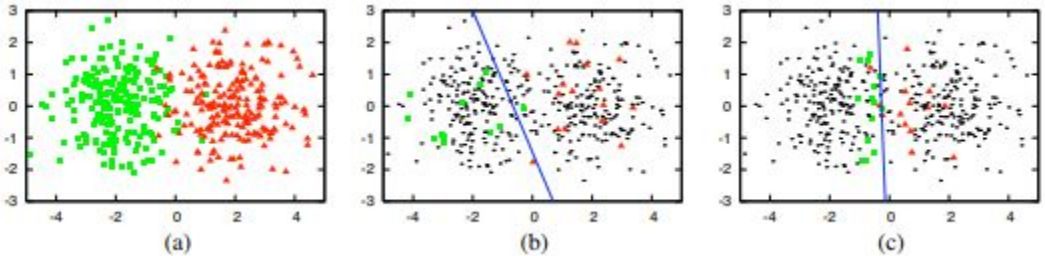


Figure 1: The pool-based active learning cycle.

Example of Active Learning



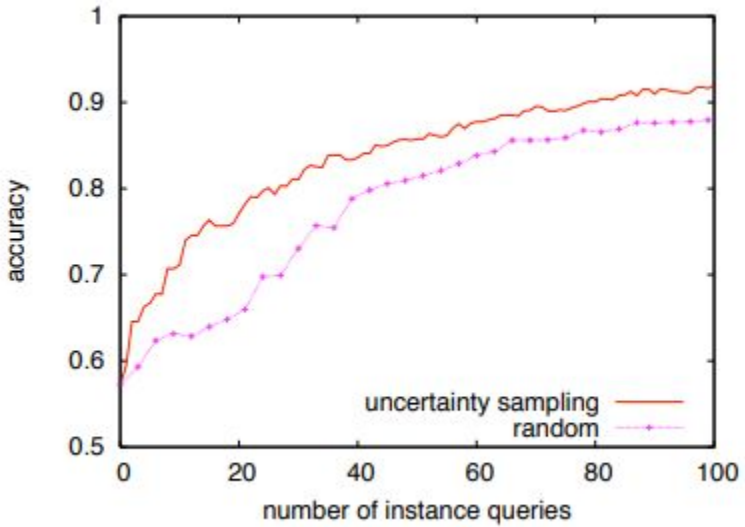
a : 두 개의 Gaussian dist 를 따르는 데이터 셋이 있다고 가정해보자.

b : Random sampling 으로, unlabeled data 에 대해 labeling을 하고, prediction (70%)

c : Query strategy(LC)에 따라, sampling 을 해서, 이에 대한 labeling 후의 prediction (90%)

아무 데이터에 대해 라벨링을 해서 decision boundary 를 그리고, prediction을 하는 것보다, decision boundary를 그릴 때, 도움이 될 즉, informative 한 unlabeled datasets을 query 하여, labeling하는 것이 더욱 효율적이다!!

Comparison between Random vs Query strategy



Active learning algorithm은 주로, learning curve로 성능 평가를 하게 된다.

decision boundary를 그리는데, 유용한 데이터(informative data)를 선별적으로 (selective) 뽑아서 라벨링하기 때문에, 학습 속도가 더욱 빨라진다.

Scenarios

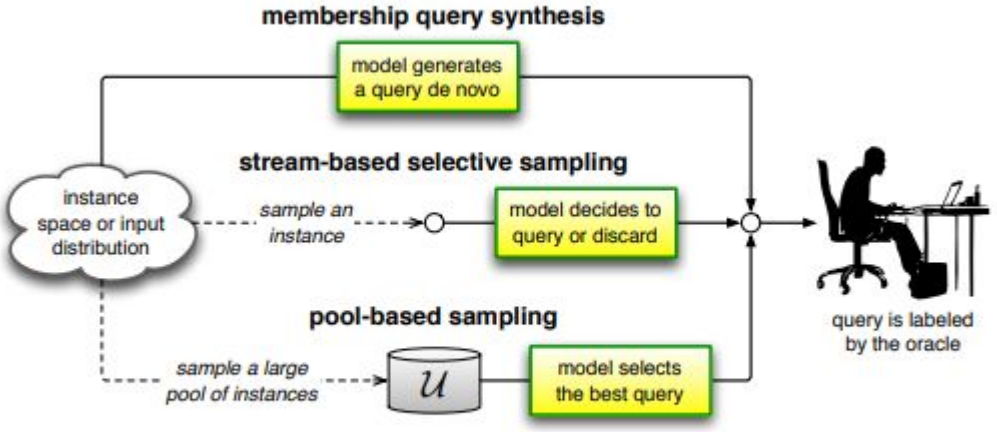


Figure 4: Diagram illustrating the three main active learning scenarios.

용어 정리 :

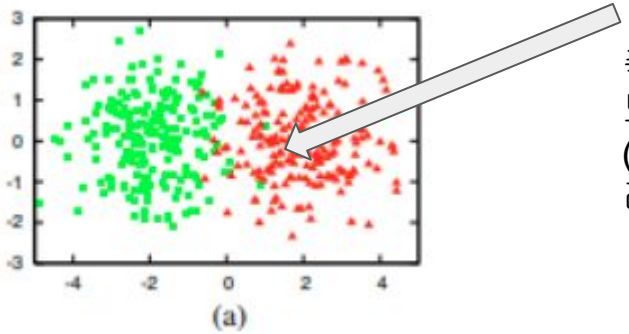
- Learner : 데이터를 학습 및 쿼리하는 객체 (모델)
- Oracle : 데이터에 레이블을 매기는 객체 (전문가)

Scenarios는 Learner 와 Oracle 사이에 datasets 의 흐름에 대한 방법론을 이야기한다.

Membership Query Synthesis

queries that the learner generates de novo, rather than those sampled from some underlying natural distribution.

Learner : “이런 데이터는 라벨을 매기기가 너무 애매해.. 화살표에 있는 데이터 [-0.1,-0.54, 0.13, 0.35 …] 는 어떤 것 같아?”



즉, 쿼리를 보낼 때, 실제로 가지고 있는 데이터에서 샘플링해서 보내는 것이 아니라, Leaner 가 input space에서 생성한 (generates de novo) 데이터를 쿼리로 보내서 Oracle에게 라벨링을 요청하는 구조.

Oracle이 human annotator일 경우, (대다수의 경우) 요청한 쿼리를 해석할 수 없다.

ex) handwritten characters

Stream-based Selective Sampling

each unlabeled instance is typically drawn one at a time from the data source, and the learner must decide whether to query or discard it.

해당 방법론의 키워드는 "draw” 와 "query strategy” 이다.

Learner : "데이터를 하나씩(sequentially) 뽑아서 보자!"

```
informative_treshold = 0.5 ; instance_inform : QueryStrategy(x)
if informative_threshold > instance_inform :
    print("query !")
else :
    print("discard!")
```

informative 에 대해 measure하는 방법론에 대해서는 다음 챕터에서 다루도록 하자!

Pool-based Sampling

evaluates and ranks the entire collection before selecting the best query.

Learner : “데이터를 보자..(unlabeled data pool)

상위 2개를 뽑기로 했으니까 Oracle한테

Data1, Data2 를 보내면 되겠다!!

Data	Informative score
Data 1	0.9
Data 2	0.4
Data 3	0.7
Data 4	0.1

Steam-base 와 Pool-base의 차이점은 전자는 sequentially 하게 개별 데이터의 정보량을 파악해 쿼리하는 것이고, Pool-base는 데이터를 모아서 pool로 보고, 이 중 정보력있는 k개를 쿼리하는 것!

Query Strategy Frameworks

Scenarios 에서 우리는, Learner 가 적은 양의 L을 통해 학습, U를 prediction하는 과정에서 informative 한 instance를 쿼리하는 일련의 구조를 살펴보았다.

query strategy는 특정 instance에 대해 얼마나 informative한지를 measure하는 다양한 방법론에 대한 이야기를 하게 된다.

- Uncertainty Sampling
  - least certain
  - margin sampling
  - entropy sampling
- Query-By-Committee
- Expected Model Change
- Expected Error Reduction
- Variance Reduction
- Density-Weighted Methods

Uncertainty Sampling

an active learner queries the instances about which it is least certain how to label. -- nearest 0.5

Least Certain $x^*_{LC} = \underset{x}{\text{argmax}}\ 1 - P_\theta(\hat{y}|x), \quad \text{where } \hat{y} = \text{argmax}_y P_\theta(y|x)$

softmax 값이 [0.2, 0.3 ,0.5] 가 하나 있고, [0.1, 0.2 ,0.7] 이 나온 두 개의 인스턴스가 있다고 하자. 각각의 y_hat 이 0.5, 0.7 이기 때문에, 더 uncertain 한 것은 첫 번째 instance !!!

Margin Sampling $x^*_M = \underset{x}{\text{argmin}}\ P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x),$

위의 예시의 경우, 각각의 margin 값은 0.5-0.3 = 0.2, 0.7 - 0.2 = 0.5 이에 따라 더 uncertain 한 것은 첫 번째 instance !!!

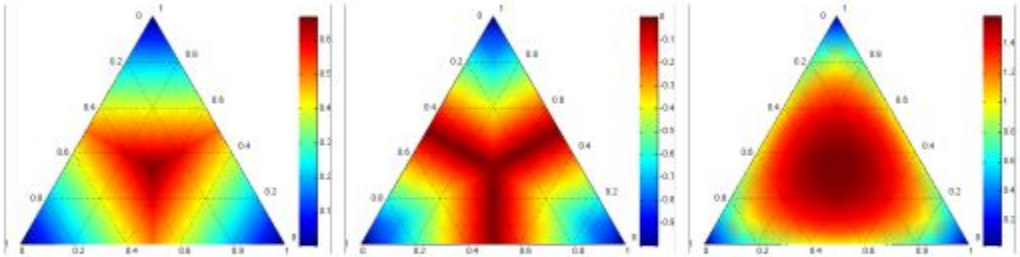
Entropy $x^*_H = \underset{x}{\text{argmax}} -\sum_i P_\theta(y_i|x) \log P_\theta(y_i|x),$

위의 예시의 경우, 각각의 entropy 값은, 각각 1.49 , 1.16 으로 더 uncertain 한 것은 첫 번째 instance !!!

```
def entropy(x) :
    entropy = 0
    for i in x :
        entropy += -1 * i * np.log2(i)
    return entropy

ls1 = [0.2, 0.3 ,0.5] ; ls2 = [0.1, 0.2 ,0.7]
entropy(ls1),entropy(ls2)
```

```
(1.4854752972273344, 1.1567796494470395)
```



entropy 방법은 하나의 라벨에 대한 probability space가 유난히 작을 경우, informative score가 낮아진다. 반면에 LC 와 margin sampling은 그 값이 높아진다.

```
ls1 = [0.01, 0.3 ,0.79] ; ls2 = [0.1, 0.2 ,0.7]
print("entropy : ",entropy(ls1),entropy(ls2))
print("LC : ",least_confidence(ls1),least_confidence(ls2))
print("Margin : ",margin_sampling(ls1),margin_sampling(ls2))
```

```
entropy :  0.8561878390097302 1.1567796494470395
LC :  0.20999999999999996 0.30000000000000004
Margin :  0.49000000000000005 0.49999999999999994
```

entropy 방법의 경우 log-loss를 최소화하는 문제에 적합하고, LC와 margin sampling의 경우, classification error를 최소화하는 문제에 적합하다. Q. 그렇다면 NLLLoss 를 사용하는 Classification의 경우에는??

Query-By-Committee

같은 데이터를 학습했지만, 다르게 학습한(다른 모델) Multiple Learner가 존재 $\mathcal{C} = \{\theta^{(1)}, \ldots, \theta^{(C)}\}$

Multiple Learner를 Committee라 하고, informative instance는 Committee 간의 prediction 차이가 가장 많이 나는 것으로 정의

Vote Entropy

$$x^*_{VE} = \underset{x}{\operatorname{argmax}} - \sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C},$$

- V(y_{i}) 는 i번째 라벨에 대해 손을 들어준 Committee의 숫자를 의미, C로 나눠줌으로써 Ratio 단위가 된다.

Kullback-Leibler (KL) divergence

$$x^*_{KL} = \underset{x}{\operatorname{argmax}} \frac{1}{C} \sum_{c=1}^{C} D(P_{\theta^{(c)}} \| P_{\mathcal{C}}),$$

두 분포의 차이를 measure

$$D(P_{\theta^{(c)}} \| P_{\mathcal{C}}) = \sum_i P_{\theta^{(c)}}(y_i|x) \log \frac{P_{\theta^{(c)}}(y_i|x)}{P_{\mathcal{C}}(y_i|x)}.$$

- P_theta(c)(y_{i}|x) 는 특정 Committee가 i번째 라벨에 대해 내린 결정 (i.e softmax)
- P_C(y_{i|}|x) 는 전체 Committee가 i 번째 라벨에 대해 내린 결정

Expected Model Change

모델을 가장 많이 변화시키는 인스턴스를 informative instance로 정의

어렸을 때, 이걸 알았다면 내 인생은 많이 변했을텐데… -> 이 인스턴스의 라벨을 안다면 내 모델의 성능은 많이 좋아졌을텐데...

“expected gradient length” (EGL) $x^*_{EGL} = \operatorname{argmax}_x \sum_i P_\theta(y_i|x) \left\| \nabla \ell_\theta(\mathcal{L} \cup \langle x, y_i \rangle) \right\|$

인스턴스의 실제 라벨을 모르기 때문에, information measure를 expectation 값으로 changed learner’s gradient 의 Euclidean 값으로 한다.

- instance의 라벨이 1이였다면, (i=1) learner의 gradient값의 Euclidean 값을, 라벨 분포로 곱해 모든 라벨을 다 더해준다.

instance에 labeling을 하기 전 즉, 기존의 Labeled instance에 대한 learner 의 gradient 는 0이라고 한다. (모델이 충분히 converge했기 때문)

이에 따라, L U <x,y_{i}> 가 아닌, <x,y_{i}> 의 gradient 만을 계산해, 연산의 효율성을 꾀한다.

$$\nabla \ell_\theta(\mathcal{L} \cup \langle x, y_i \rangle) \approx \nabla \ell_\theta(\langle x, y_i \rangle)$$

Expected Error Reduction

이전의 방법론은 특정 인스턴스가 모델을 얼마나 변화시킬 지에 집중했다면, 해당 방법론은 해당 인스턴스를 사용함으로써, 모델의 에러가 얼마나 감소할 지에 집중한다.

보다 구체적으로 이야기하자면, unlabeled instance에서 특정 instance를 쿼리한 후, retraining 한 모델이 remaining unlabeled instance를 prediction했을 때, 최대한 certain 해져야 한다.

0-1 loss

$$x^*_{0/1} = \underset{x}{\operatorname{argmin}} \sum_i P_\theta(y_i|x) \left(\sum_{u=1}^{U} 1 - P_{\theta^{+\langle x, y_i \rangle}}(\hat{y}|x^{(u)}) \right)$$

log loss (entropy)

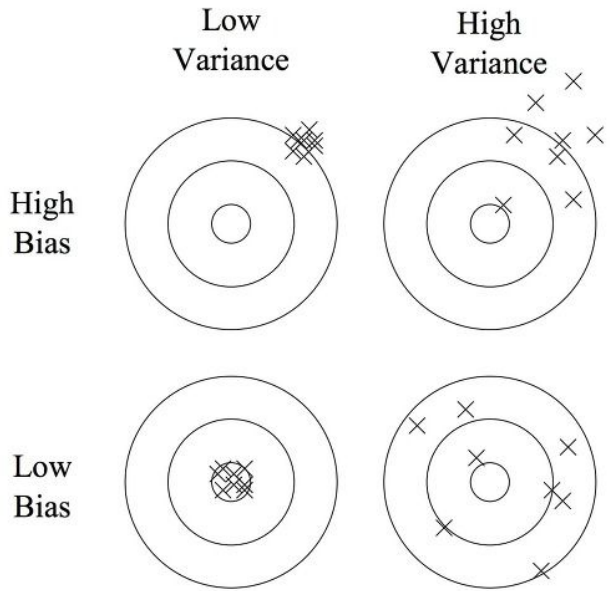
$$x^*_{\log} = \underset{x}{\operatorname{argmin}} \sum_i P_\theta(y_i|x) \left(-\sum_{u=1}^{U} \sum_j P_{\theta^{+\langle x, y_i \rangle}}(y_j|x^{(u)}) \log P_{\theta^{+\langle x, y_i \rangle}}(y_j|x^{(u)}) \right)$$

0-1 loss 만 봐도, sigma 가 라벨(i)에 대해서, unlabeled(U)에 대해서 존재한다.
이 때, U가 매우 크기 때문에, computational burden 이 매우 큰 방법론이다.

Variance Reduction

모델의 loss function 를 최소화하는 문제는 비용도 많이 들고, 닫힌 해가 아닐 수 있다.

그렇다면, 모델의 Variance 를 줄여보는 인스턴스를 찾는 것은 어떨까?



Error(X) = noise(X) + bias(X) + variance(X)

noise 는 데이터가 가지는 본질적인 한계치 -- irreducible

Bias 는 모델이 잘 못맞추는 것이다. 모델이 별로거나, 데이터가 적거나의 문제이다. (underfitting) -- irreducible

Variance 너무 잘 맞추려다 보니, 그것만 잘 맞추게 되는 민감도의 문제이다. (overfitting) -- reducible

Variance Reduction

$$x^*_{VR} = \underset{x}{\operatorname{argmin}} \ \langle \tilde{\sigma}^2_{\hat{y}} \rangle^{+x}$$

neural networks 의 output variance's approximation (MacKay, 1992)

$$\sigma^2_{\hat{y}}(x) \approx \left[\frac{\partial \hat{y}}{\partial \theta}\right]^{\mathsf{T}} \left[\frac{\partial^2}{\partial \theta^2} S_\theta(\mathcal{L})\right]^{-1} \left[\frac{\partial \hat{y}}{\partial \theta}\right] \approx \nabla x^{\mathsf{T}} F^{-1} \nabla x$$

우변의 양 끝은 prediction 이 parameter에 대해 얼마나 민감한 지에 대한

우변의 가운데 term은 현재 모델의 squared error이다.(L로 학습한) 해당 부분을 Fisher information matrix라 하고, F로 정의 (cov of LL)

- 이상적인 경우, loss function 은 convex 하기 때문에, gradient of x 는 locally linear 하고, variance 형태인 F는 Gaussian 을 따르게 되기 때문에, Closed form 이 된다. 이에 따라, 모델의 re-train 이 불필요하게 된다.
  - closed form : 무한대(infinity)로 발산하거나 조건부적인 결과를 제공하는 경우가 아닌 경우

Fisher information 은 random variable X를 모델링하는 매개 변수 θ에 따라 발생되는 정보량을 의미한다.

output variance에서 F의 inverse 가 들어가게 되고, 사실상 Fisher information 은 maximizing 해야 한다. -> 모델의 정보량을 최대화

결과적으로, 모델의 variance 를 크게 만드는 instance 를 라벨링함으로써 모델의 variance를 줄여주는 instance x를 찾는 방법론

Density-Weighted Methods

오른쪽 plot을 보면, decision boundary 형성에 중요한 데이터는 색이 있는

instance지만, A가 decision boundary 위에 존재, 가장 informative하게 된다.

즉, informative 할 수는 있지만, 전체 input space에 대한 대표성(representativ

없다고 할 수 있다.

이에 따라, query하려는 instance와 U에 있는 remaining instance간의 similarity를 계산하고, 이를 다른 query strategy와 가중곱을 해주어 query instance가 다른 U 데이터들에 대해 대표성을 띄는 것을 보장해준다.

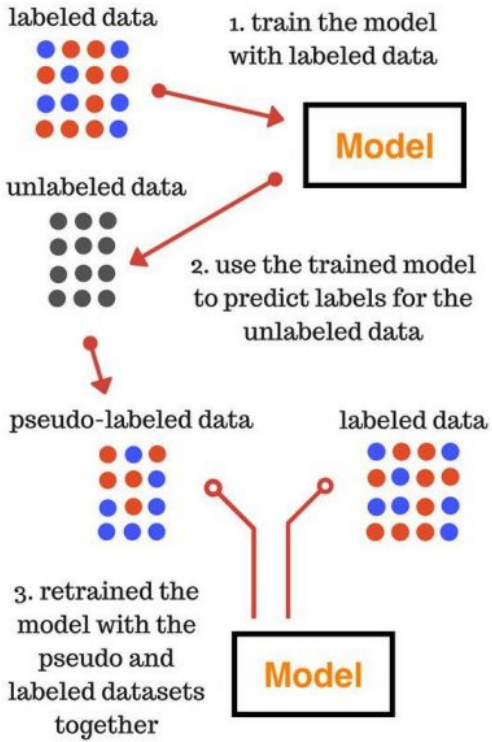
$$x^*_{ID} = \underset{x}{\operatorname{argmax}}\, \phi_A(x) \times \left(\frac{1}{U} \sum_{u=1}^{U} \operatorname{sim}(x, x^{(u)}) \right)^{\beta}$$

Related Research Areas

- semi-supervised learning :

unlabeled data로 self-training 을 한 후에, supervised learning을 진행하는 과정이 active learning 과 유사한 관계에 있다.

다만 semi-supervised learning은 적은 양의 L로 학습한 모델로 most confident 한 label을 L set 으로 추가시켜 re training 을 시키는 구조이지만, least confident 한 instance 를 query 하는 active learning과는 complement 한 관계성을 가지고 있다고 할 수 있다.

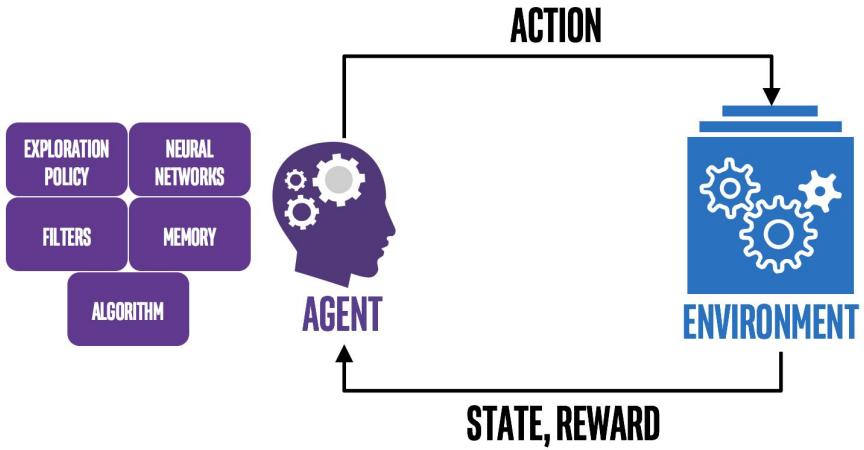


Related Research Areas

- reinforcement learning :

강화학습에서 learner는 "행동"을 통해 주어진 환경과 상호 작용하고 환경으로부터받는 "보상"과 관련하여 행동의 최적 정책을 찾으려고 한다.

"행동" 에 따른 결과를 query strategy 에 따른 informative score라고 하고, 이에 따른 oracle의 labeling을 "보상" 과 같은 응답 체계로 치환하게 되면, active learning과 유사한 구조를 띄게 된다.



Additional Discussion

- **Deep Active Learning for Text Classification**
  - Bang An , Wenjun Wu , Huimin Han

- **Active Discriminative Text Representation Learning**
  - Ye Zhang , Matthew Lease , Byron C.

- **AILA: Attentive Interactive Labeling Assistant for Document Classification through Attention-based Deep Neural Networks**
  - Minsuk Choi 1 , Cheonbok Park 1 , Soyoung Yang 1 , Yonggyu Kim 2 , Jaegul Choo 1 , and Sungsoo (Ray) Hong

Deep Active Learning for Text Classification

Text classification 에는 RNN based neural network 를 stack 한 형식을 사용하였다.

Query strategy 는 least confident 방법을 사용하되, softmax 값의 평균값에서 얼마나 벗어났는지를 측정하였다. (아래의 공식)

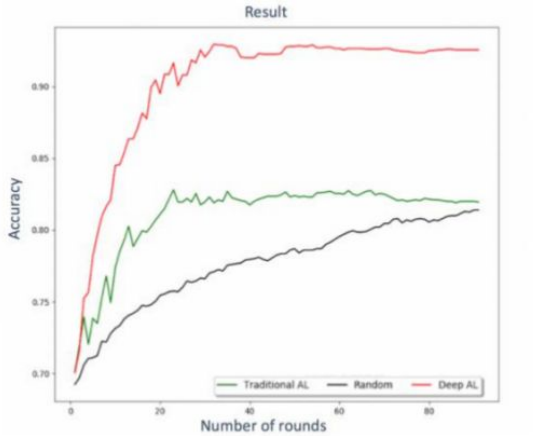
$$x^*_{Softmax} = P_\theta(\hat{y}|x) - P_\theta(\bar{y}|x)$$

traditional AL 은 instance 를 sequential 하게 보거나, unlabeled pool 을 보고, k 개를 query 하는 방식을 사용하였다 .high computational expense !!

즉, stream base 는 속도가 너무 느리고, pool base 는 memory cost 가 너무 큰 단점이 존재한다.

이에 따라, batch mode 로 instance 를 query 하는 방법론인 batch-model for active learning 을 사용,

학습의 속도와 computational expense 를 줄이는 효과를 보인다.



red : with batch
green : traditional AL
grey : random sampling

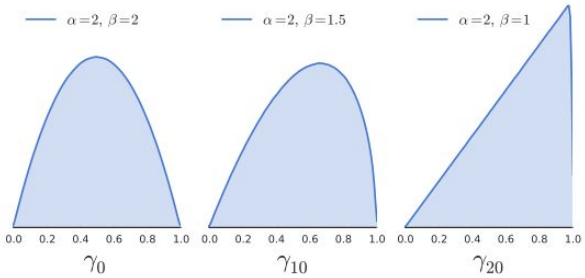
Active Discriminant Text Representation Learning

text classification 에 CNN architecture를 사용

모델의 lowest layer인 embedding layer 의 gradient 를 가장 많이 변화시키는 instance를 query하는 EGL approach 사용 (왼쪽의 첫 번째 formula)

긴 문장에 대해서 성능을 높이기 위해, (1) gradient (2) uncertainty 를 모두 고려할 수 있는 EGL-entropy-beta model 제안 (왼쪽의 두 번째 formula) 가중치 lambda 는 iteration 이 높아질 수록, 두 번째 term이 smooth 하게 커지게 하는 beta dist 를 따름

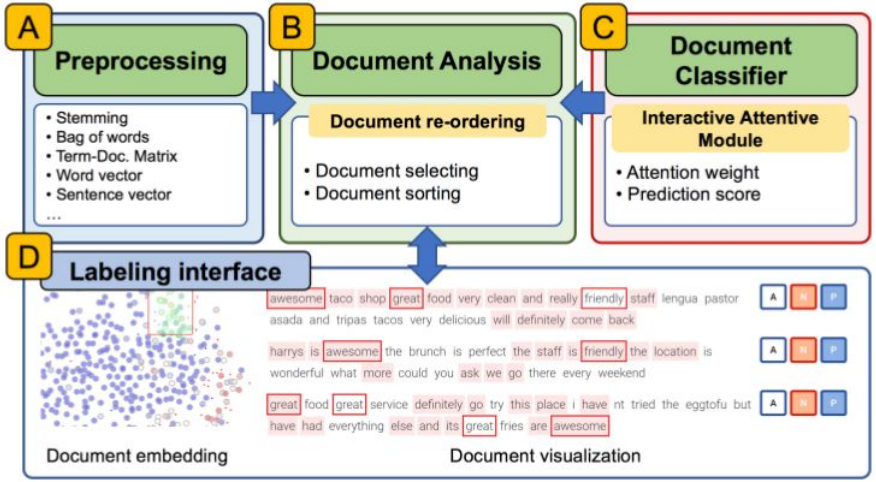
$$\max_{j \in \mathbf{x}_i} \sum_k P(y_i = k|\mathbf{x}_i; \boldsymbol{\theta}) \|\nabla J_{\mathbf{E}^{(j)}}(\langle \mathbf{x}_i, y_i = k \rangle; \boldsymbol{\theta})\|$$



$$\phi_t(i) = \gamma_t \cdot \mathcal{P}(\phi_{\text{Entropy}}(i), \{\phi_{\text{Entropy}}(j) : j \in \mathcal{U}\}) + (1 - \gamma_t) \cdot \mathcal{P}(\phi_{\text{EGL-word-doc}}(i), \{\phi_{\text{EGL-word-doc}}(j) : j \in \mathcal{U}\})$$

AILA

기존의 AL은 Oracle 이 라벨만을 부여했지만, 해당 모델에서는 Attention 을 할 단어나 부분을 annotate 해주게 된다. 따라서, loss function이 label 뿐만 아니라, attention 에 대해서도 적용이 되는 구조를 가지게 되고, 이에 따라, query 도 label 과 attention 에 대해 두 가지를 하게 된다.



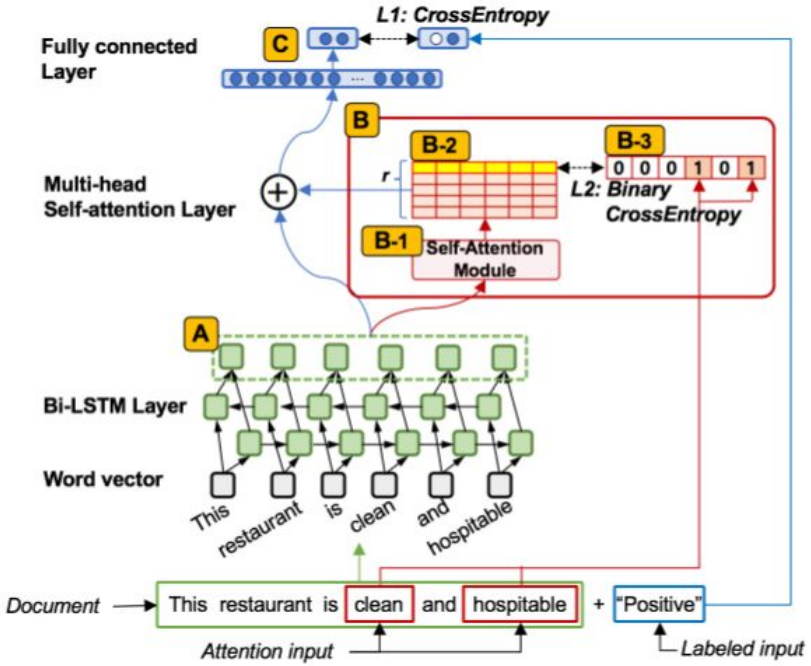


Figure 2: Overview of our model: (A) word vector conversion, (B) IAM, and (C) the classifier.

Further Study

- Batch model Active Learning
  - Batch Mode Active Learning and Its Application to Medical Image Classification (2006), Steven C. H. Hoi et.al

- Optimal size of Labeled data

- etc ..

Thank You