

Synthesized Classifiers for Zero-Shot Learning

Método SYNC

Dadas as descrições semânticas das classes dos objetos, o ZSL visa reconhecer com precisão objetos de classes desconhecidas (cujos exemplos não estão disponíveis na fase de treino) associando-os às classes conhecidas (cujos exemplos etiquetados estão disponíveis).

A ideia deste método é alinhar o espaço semântico, que deriva de informação externa, com o espaço do modelo, que se preocupa com o reconhecimento visual de características. Para este fim, foi introduzido um conjunto de classes "*phantom*" cujas coordenadas estão presentes em ambos os espaços: semântico e modelo. Servindo como bases num dicionário, podem ser otimizados a partir de dados etiquetados de forma a que os classificadores de objetos reais sintetizados atinjam a performance discriminativa ideal.


Introdução

- **(1): Como relacionar as classes desconhecidas com as classes conhecidas?**

Para endereçar esta questão, os investigadores têm usado atributos visuais e *word vectors*, para associar classes desconhecidas e conhecidas. São os chamados *semantic embeddings* dos objetos.

- **(2): Como alcançar a performance ótima nas classes desconhecidas mesmo que nós não tenhamos esses dados etiquetados?**

O desenho de modelos probabilísticos e os *nearest neighbor classifiers* no espaço semântico são soluções a considerar. Finalmente, classificadores para as classes desconhecidas podem ser construídos diretamente no *input* do espaço de características.

 Ilustração do Método para ZSL

Tal como pode ser observado pela imagem, as classes dos objetos estão em ambos os espaços. As classes são caracterizadas no espaço semântico com *semantic embeddings* (a_i), como por exemplo atributos e vetores de palavras. São também representadas como modelos para reconhecimento visual (w_i) no espaço do modelo. Em ambos os espaços, estas classes formam um grafo pesado. A ideia base é que ambos os espaços estejam alinhados. Em particular, as coordenadas no espaço do modelo devem ser a projeção do grafo de vértices do espaço semântico para o espaço do modelo, preservando a relação codificada no grafo. São introduzidas as classes "*phantom*" (b e v) para ligar

as classes conhecidas e desconhecidas. Os classificadores para as classes "*phantom*" são bases para os "*synthesized classifiers*" para as classes reais. Em particular, a síntese toma a forma de uma combinação convexa.

Como é alinhado o espaço semântico com o espaço do modelo?

As coordenadas do espaço semântico dos objetos são designadas ou derivam de informação externa (como dados textuais), enquanto que o espaço do modelo se preocupa com o reconhecimento de características visuais de baixo nível.

Para alinhar ambos os espaços, as coordenadas do espaço do modelo são vistas como a projeção dos vértices do grafo do espaço semântico, por exemplo, seguindo a abordagem do algoritmo Laplacian eigenmaps.

Para adaptar as embeddings aos dados, foram introduzidas um conjunto de classes *phantom* - as coordenadas destas classes em ambos os espaços são ajustáveis e otimizadas de forma a que o modelo resultante para as classes dos objetos reais atinjam a melhor performance nas tarefas discriminativas.

Os modelos para as classes *phantom* podem ser interpretados como classificadores num dicionário onde uma grande quantidade de classificadores para as classes reais pode ser sintetizado através de combinações convexas.

Quando nós precisamos de construir um classificador para uma classe desconhecida, nós calculamos os coeficientes de combinação convexa das coordenadas da classe no espaço semântico e usamos o resultado para contruir o classificador correspondente.

A principal contribuição é uma nova ideia para resolver o problema de reconhecer classes desconhecidas a partir aprendizagem de inúmeros embeddings de grafos compostos por classes de objetos.

Nota: Um **embedding** é a tradução de um espaço de elevada dimensão para um espaço de baixa dimensão. Os *embeddings* tornam a tarefa de *machine learning* mais fácil, onde existem muitos *inputs*, como por exemplo vetores esparsos que representam palavras. Idealmente, um *embedding* captura alguma informação semântica do input colocando inputs semânticos similares todos juntos no espaço *embedding*. Um *embedding* pode ser aprendido e reutilizado ao longo dos modelos.

An embedding is a matrix in which each column is the vector that corresponds to an item in your vocabulary. To get the dense vector for a single vocabulary item, you retrieve the column corresponding to that item. (*Machine Learning Crash Course, Google*)

Related Work

A fim de transferir conhecimento entre classes, o ZSL depende de embeddings semânticas das etiquetas de classes, incluindo atributos, vetores de palavras, conhecimento extraído da Web ou a combinação de vários embeddings.

Dadas as embeddings semânticas, as abordagens existentes ZSL caem normalmente em métodos baseados em embeddings e similaridades.

Nas abordagens baseadas em embeddings, primeiro a imagem de input é mapeada no espaço semântico e depois é determinada a etiqueta da classe nesse espaço por várias medidas de relacionamento implícitas pelas embeddings de classes.

Já nas abordagens baseadas na similaridade, primeiro são construídos classificadores para as classes desconhecidas relacionando-os com as classes conhecidas através da similaridade entre classes.

A abordagem apresentada neste paper partilha do espírito destes modelos mas oferece uma flexibilidade de modelação mais rica graças à introdução das classes *phantom*.

O método apresentado não tem acesso a dados das classes desconhecidas.

Abordagem

- Cada classe c tem uma coordenada a_c no espaço de embeddings semânticas;
- Adicionalmente, foram introduzidas um conjunto de classes *phantom* associadas a embeddings semânticas b_r ;
- As classes *phantom* e reais formam um grafo pesado biparticional com os pesos definidos como:

$$s_{cr} = \frac{\exp\{-d(a_c, b_r)\}}{\sum_{r=1}^R \exp\{-d(a_c, b_r)\}}$$

para correlacionar a classe real c e a classe *phantom* r , onde

$$d(a_c, b_r) = (a_c - b_r)^T \sum_{i=1}^{-1} (a_c - b_r)$$

e \sum é um parâmetro que pode ser aprendido a partir dos dados, modelando a correlação entre os atributos. ($\sum = \sigma^2 I$).

- A forma específica de definir os pesos é motivada por vários métodos de aprendizagem como por exemplo o *Stochastic Neighbor Embedding* (SNE). Em particular, s_{cr} pode ser interpretado como a probabilidade condicional de observar a classe r nas vizinhanças da classe c .

- No espaço do modelo, cada classe real é associada ao classificador w_c e a classe *phantom* r é associada ao classificador virtual v_r .
- O alinhamento dos espaços semântico e modelo é feito vendo w_c (ou v_r) como um embedding de um grafo pesado.
- Em particular, foi considerada a ideia por detrás dos Laplacian eigenmaps, que procuram que as embeddings mantenham a estrutura do grafo tanto quanto possível.
- A fórmula para síntese dos classificadores é a seguinte:

$$w_c = \sum_{r=1}^R s_{cr} v_r$$

Por outras palavras, a solução dá a ideia de classificadores sintetizados dos classificadores virtuais v_r .

O problema de aprendizagem é aprender as coordenadas *phantom* v e b para um desempenho ótimo de discriminação e generalização.

Aprender classes *phantom*

Aprender classificadores base

Os classificadores base $\{v_r\}_{r=1}^R$ são aprendidos através dos dados de treino (apenas das classes conhecidas).

- Classificadores One-vs-Rest;
- Crammer-Singer multi-class SVM loss;

Aprender embeddings semânticas

A equação do grafo pesado é parametrizada por embeddings adaptadas de classes *phantom* b_r . Neste trabalho, para simplificação, assumimos que cada embedding é uma combinação linear esparsa dos vetores de atributos das classes conhecidas.

$$b_r = \sum_{c=1}^S \beta_{rc} a_c$$

Comparação com vários métodos existentes

- O COSTA^[1] combina classificadores pré-treinados de classes conhecidas para construir novos classificadores. Para estimar o embedding semântico (word vector) de uma imagem de teste, usa valores de decisão de classificadores pré-treinados de objetos já vistos para a média ponderada da embedding semântica correspondente.

Experiências

- Foram utilizados os *datasets* **AWA**, **CUB**, **SUN** e **ImageNet**;

Espaços Semânticos

- Nenhum dos *datasets* tem vetores de palavras para os nomes das classes, por isso foram obtidos através de técnicas^[2] de extração de vetores de palavras.
- No caso do ImageNet foi treinado um modelo de linguagem skip-gram^[3] para extrair vetores de palavras de 500 dimensões para cada classe.

Caraterísticas Visuais

- Foram extraídas caraterísticas recorrendo à AlexNet para o AWA e para o CUB, e ao GoogLeNet para todos os *datasets*.

Protocolos de avaliação

- Para o AWA, CUB e SUN, foi usada a accuracy *multi-way classification*. No caso do ImageNet foram usadas duas métricas de avaliação: F@K e HP@K [13].
- O F@K é definido como a percentagem de imagens de teste que o modelo considerou como etiquetas positivas no top K de previsões.
- O HP@K tem em conta a organização hierarquica das categorias dos objetos. Para cada etiqueta positiva, é gerada uma lista de grau de confiança das K categorias mais próximas na hierarquia e calculado o grau de sobreposição (precisão) entre o grau de confiança e as K top previsões do modelo.
- Os métodos foram avaliados em três cenários de dificuldade incremental.

Detalhes da Implementação

Por conveniência, foi definido o número de classes *phantom* como sendo o mesmo que o número de classes conhecidas e $b_r = a_c$ para $r = c$.

Os métodos usados são convencionados da seguinte maneira:

- $Ours^{o-vs-o}$: one-versus-other
- $Ours^{cs}$: Crammer Singer
- $Ours^{struct}$: Crammer Singer with structured loss

Resultados Experimentais

No caso de datasets de larga-escala, como o ImageNet, os resultados foram comparados ao método ConSE, que é o melhor método estado da arte neste dataset, à data do artigo. No entanto, o método

apresentado neste paper supera os resultados do ConSE.

Vantagem dos atributos contínuos

- Os atributos contínuos (fornecem um valor mais preciso) como embeddings semânticas para classes obtêm melhor performance do que atributos binários. Isto é especialmente verídico quando são usadas características profundas para a construção de classificadores.

Vantagem das características profundas

- É também conclusivo que as características profundas melhoram significativamente a performance. A GoogLeNet supera geralmente a AlexNet.

Que tipos de espaços semânticos

- Foi descoberto que os atributos conseguem melhor performance do que os vetores de palavras;
- No entanto, quando combinados os dois, obtém-se ainda melhores resultados;

Quantos classificadores base são necessários?

- O estudo demonstra que um número próximo dos 60, 70% em relação ao número de classes conhecidas é suficiente para atingir o plateau da curva de performance. Um número maior não causa grande efeito.

Conclusão

Foi desenvolvido um novo mecanismo de síntese do classificador para ZSL com a introdução da noção de classes *phantom*.

As classes *phantom* ligam os pontos entre as classes conhecidas e desconhecidas - os classificadores das classes conhecidas e desconhecidas são construídos a partir dos mesmos classificadores base para as classes *phantom* e com as mesmas funções de coeficientes. Como resultado, podemos aprender convenientemente o mecanismo de síntese do classificador aproveitando os dados das classes conhecidas e em seguida aplicá-los prontamente às classes desconhecidas.

1. [COSTA](#) ↩
2. Ver os artigos [14, 15]; ↩
3. O Skip-gram é uma das técnicas de aprendizagem não supervisionada usada para encontrar as palavras mais relacionadas para uma dada palavra. ↩