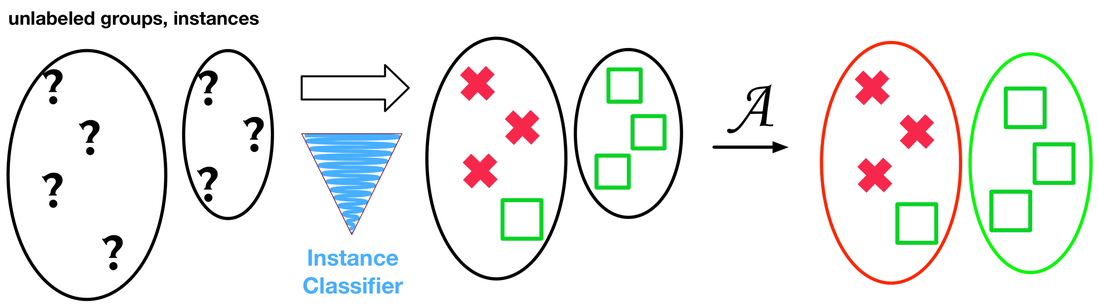
From Group to Instance Labels, using Deep Learning

Dimitrios Kotzias[1,2] Misha Denil[2] Nando de Freitas[2,3] Padhraic Smyth[1]

[1]University of California, Irvine [2]University of Oxford [3]CIFAR
{dkotzias,smyth}@ics.uci.edu {misha.denil,nando}@cs.ox.ac.uk

Motivation

- We often have labels for *groups of instances*, but not each individual one.
- We want to learn a classifier for both unseen groups and instances
- We achieve that by building an instance classifier, based on instance similarity and group constrains



We demonstrate this idea by inferring the ratings of sentences (individuals) from ratings of reviews (groups)

Typical Application Example

31 out of 45 people found the following review useful:

Beautifully acted, but both leading and misleading

Author: siderite from Romania

28 January 2010

Paul Bettany did a great role as the tortured father whose favorite little girl dies tragically of disease. For that, he deserves all the credit. However, the movie was mostly about exactly that, keeping the adventures of Darwin as he gathered data for his theories as incomplete stories told to children and skipping completely the disputes regarding his ideas.

Two things bothered me terribly: the soundtrack, with its whiny sound, practically shoving sadness down the throat of the viewer, and the movie trailer, showing some beautiful sceneries, the theological musings of him and his wife and the enthusiasm of his best friends as they prepare for a battle against blind faith, thus misrepresenting the movie completely.

To put it bluntly, if one were to remove the scenes of the movie trailer from the movie, the result would be a non descript family drama about a little child dying and the hardships of her parents as a result. Clearly, not what I expected from a movie about Darwin, albeit the movie was beautifully interpreted.

A positive review, with both positive and negative sentences.

Multi-Instance Learning Applications

- Previous work in MIL typically learns (Kueck et al., 2004) or assumes a specific aggregation function $\mathcal{A}$ (**OR:** Dietterich et al., 1997, **Average:** Xu et al., 2004)
- $\mathcal{A}$ maps labels of instances to the group label.
- Applications include: Review Classification, Image Recognition, Privacy, Comparative analysis (UI)

Proposed Cost Function

Cost Function is based on the fact that:

- Similar instances should have a similar score
- The label for a group G should be a function of the label's of the group's instances

and hence consists of two parts:

$$J(\theta) = \textit{Instance Cost} + \textit{Group Cost}$$

$$J(\theta) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j)\Delta_1\left(\hat{y}_i,\hat{y}_j\right) + \lambda\frac{1}{K}\sum_{k=1}^{K}\Delta_2\left(\hat{\ell}_k,\ell_k\right)$$

- $\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j) \in [0,1]$ similarity measure between instances $\mathbf{x}_i,\mathbf{x}_j$;
- $\hat{y}_\theta(\mathbf{x}_i)$ prediction for instance $\mathbf{x}_i$
- Δ_1,Δ_2 penalty for the difference between their arguments
- $\hat{\ell}_k = \mathcal{A}\left(\mathcal{G}_k,\theta\right) \in [0,1]$ is the prediction for group k.
- $\lambda > 0$ balances the contributions between the 2 costs

Overall Approach

- Create representation of instances (here we represent sentences as real-valued vectors)
- Optimize cost function
- Use mini-batch gradient descent to avoid $\mathcal{O}(N^2)$ complexity
- Learn λ through linear search
- Evaluation is measured on previously unseen groups and instances

Specific Choices for our Experiments

- $\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\|\mathbf{x}_i-\mathbf{x}_j\|_2^2)$
- $\Delta_1 = \Delta_2 = \Delta(\hat{y}_i,\hat{y}_j) = (\hat{y}_i-\hat{y}_j)^2$

$$\hat{y}_i = \hat{y}_\theta(\mathbf{x}_i) = \sigma(\theta^\top\mathbf{x}_i) = \frac{1}{1+e^{-\theta^\top\mathbf{x}_i}}$$

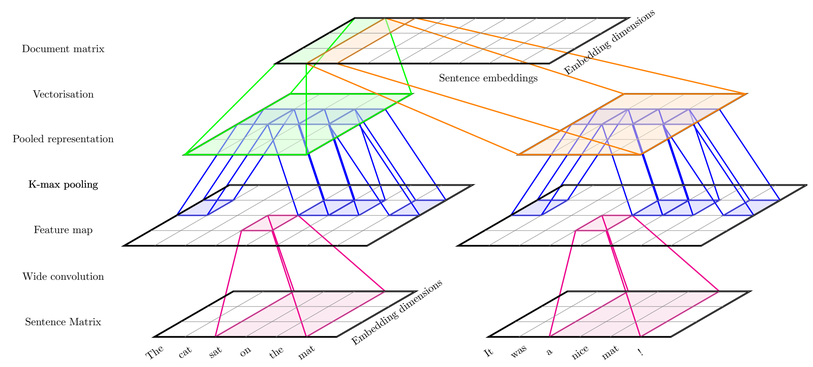
$$\hat{\ell}_k = \mathcal{A}(\mathcal{G}_k,\theta) = \frac{1}{|\mathcal{G}_k|}\sum_{i\in\mathcal{G}_k}\hat{y}_\theta(\mathbf{x}_i)$$

Deep NLP for Feature Learning

BOW is not a good similarity measure for sentences.
Use distributed representations of words *word* $\rightarrow \mathbf{x} \in \mathcal{R}^n$
Larger blocks of text, paragraphs, documents (Le et al., 2014, Denil et al., 2014)

$$\textit{sentence} \rightarrow \mathbf{x} \in \mathcal{R}^n$$

We train the convolutional network for documents of Denil et al (2014), which only requires labels for documents, but is able to generate features for words, sentences and the documents.



Datasets

- 50,000 IMDb movie reviews
  Maas et al 2011
- 70,000 Amazon reviews
  McAuley and Leskovec 2013
- 600,000 Yelp reviews
  Yelp Dataset Challenge 2015

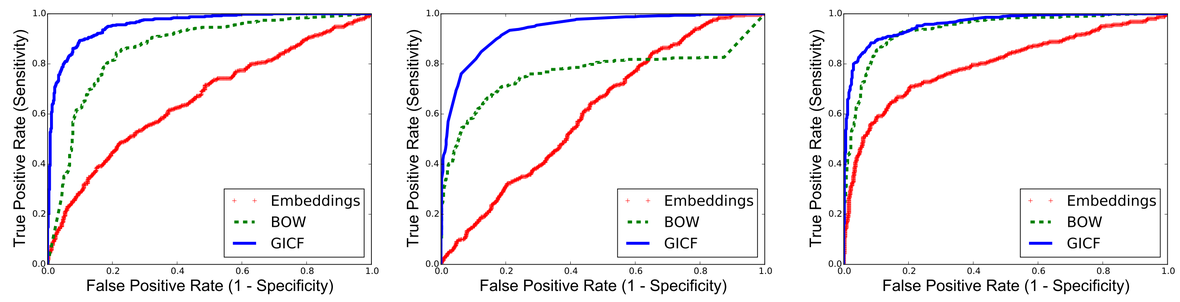
Binary Labels from review scores

Instance Evaluation

Hand-labelled 1000 sentences from each dataset to evaluate.

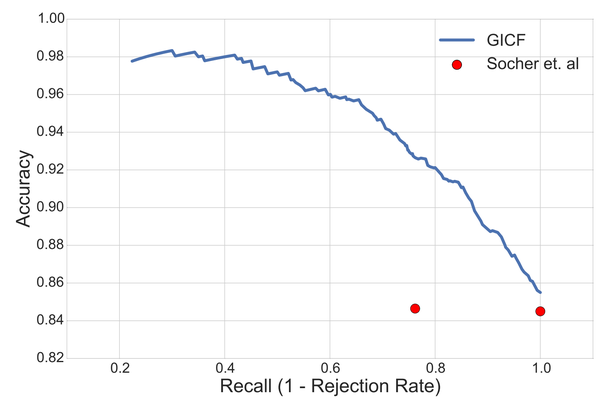
Baselines used:

- logistic regression on BOW
- logistic regression on embeddings
- *Socher et al., 2013* method for movie comparison



(a) Amazon sentence ranking (b) Imdb sentence ranking (c) Yelp Sentence ranking

ROC plots for instance level classification, for each of the baselines and our method for the three datasets



Accuracy of sentence classification, for various levels of rejection rate (neutral sentences) in **IMDb** dataset. We set a boundary b around the 0.5 point, to create a neutral class

Group Evaluation

	Accuracy			AUC		
	Amazon	IMDb	Yelp	Amazon	IMDb	Yelp
Logistic w/ BOW	85.8%	86.20%	**91.25%**	88.08%	88.32	**94.41**
Logistic w/ embeddings	67.82%	58.23%	81.00%	61.24%	60.77	82.59
GICF w/ embeddings	**92.8%**	**88.56%**	88.73 %	**91.73%**	**88.36%**	92.36%

Table : Accuracy and Area-Under-the-Curve (AUC) scores for predicting labels at the group (document) level for the baselines and our proposed method (GICF).