Weakly Supervised Discriminative Training of Linear Models for Natural Language Processing

Lina Maria Rojas Barahona
Christophe Cerisara

lina.rojas@eng.cam.ac.uk
cerisara@loria.fr

CNRS / LORIA UMR 7503
Vandoeuvre-les-Nancy, France

Introduction

Importance of unsupervised training for NLP:

- Avoid costly manual annotations for every new task/domain/language
- Language is in permanent evolution (Fromreide,2014): must annotate again and again...
- Era of Big data: too much data to annotate
- Avoid vanishing gradient in deep learning

Introduction

Main challenge: Very hard to use the same model & ”error” objective as used at test time:

State-of-the-art solutions

- Common: *generative model* of observations $\neq$ classification
  e.g. RBM in deep networks, clustering, Bayesian networks...
- *discriminative model* of observations
  e.g. Word2Vec, autoencoders in deep learning...

But not optimum with regard to classification error at test time !

Introduction

Empirical classifier error = approx. of classifier *risk*:

$$R(\theta) = E_{p(X,Y)}\mathcal{L}(Y, f_\theta(X)) \simeq \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(Y^{(i)}, f_\theta(X^{(i)}))$$

Proposed solution in (Balasubramanian, JMLR 2011)

New approximation of classifier risk:

- Numeric computation of the expectation (integral)
- Assumes priors $p(Y)$ known
- Assumes $p(f_\theta(X)|y)$ is Gaussian
- Only works for binary linear classifiers

Introduction

Main idea from (Balasubramanian, JMLR 2011):

$$\begin{aligned} R(\theta) &= E_{p(X,Y)}\mathcal{L}(Y, f_\theta(X)) \\ &= \sum_{y\in\{0,1\}} P(y) \int_{-\infty}^{+\infty} P(f_\theta(X) = \alpha | y)\mathcal{L}(y, \alpha) d\alpha \end{aligned}$$

General algorithm

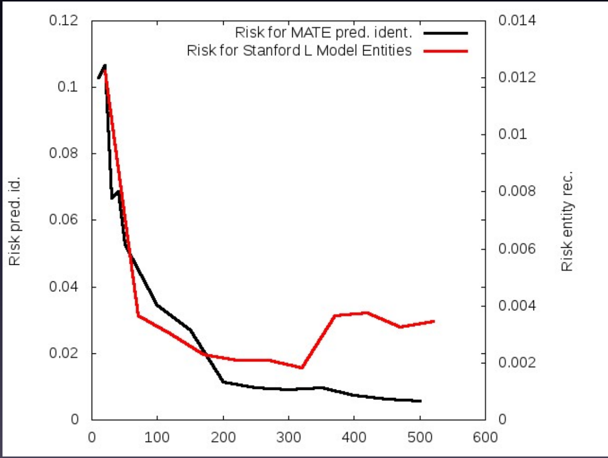
1. Start from random linear weights
2. Compute linear scores $f_\theta(X)$ on unlabeled corpus
3. Cluster scores into 2 Gaussians with EM
4. Numerical integration $\rightarrow \hat{R}(\theta)$
5. Finite difference $\rightarrow \nabla\hat{R}(\theta)$
6. Gradient descent + iteration from 2)

Contributions

- Derivation of *closed-form expectation* (no more num. int.)
  - (see paper & additional material for details)
- Improved convergence: weakly supervised init. of weights
- Study on 2 NLP tasks:
  - Predicate identification (binary: predicate or not)
    - Europarl CLASSIC corpus: 1000 sentences in French
    - 10 annotated sentences to initialize the weights
    - assumed prior: 20% of predicates
    - Same features as MATE SRL: POS-tags, dep. relations
  - Entity detection (binary linear classifier: entity or not)
    - ESTER2 broadcast news French corpus
    - 20 annotated sentences to initialize the weights
    - Same features as Sanford NLP: POS-tags, letter 4-grams, capitalization...
    - assumed prior: 10% of entities

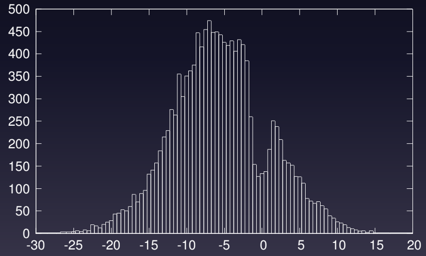
Validation of the Risk

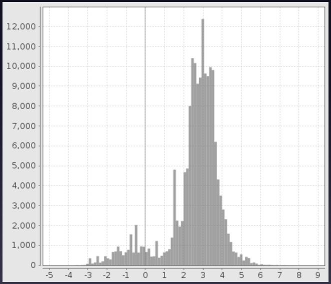
Is this risk estimation related to task classification error ?
Evolution of the risk of supervised classifiers with training corpus size:



Gaussianity assumption

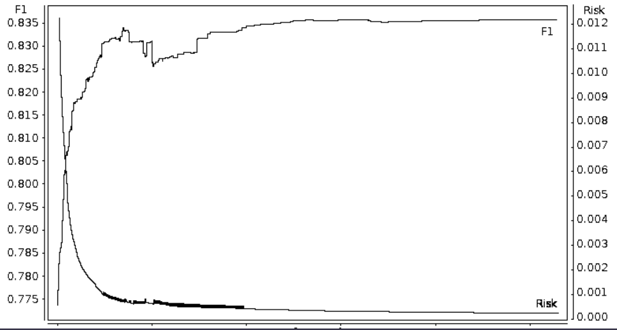
Distribution of the linear scores during optimization for predicate identification (left) and entity recognition (right):





Convergence of the risk

Entity detection, unsupervised iterations:



Evaluation on the first task

Classifier performances on predicate identification:

Task 1			
System	F1	precision	recall
MATE trained on 10 sent.	64.8%	72.1%	58.9%
MATE trained on 500 sent.	87.2%	92.0%	82.9%
Weakly supervised	**73.1**%	63.1%	**87.1**%

- Comparable to supervised classifier trained on several hundreds sentences
- Best recall

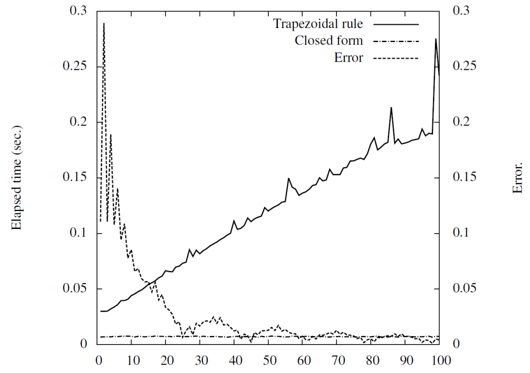
Evaluation on the first task

Classifier performances on entity recognition:

Task 2			
System	F1	precision	recall
Stanford trained on 20 sent.	77.4%	89.8%	68%
Stanford trained on 520 sent.	87.5%	90.3%	84.7%
Weakly sup. closed-form risk	**83.5**%	88.9%	**78.7**%
Weakly sup. numerical integration	**83.6**%	88.7%	**79**%

- Comparable to supervised classifier trained on several hundreds sentences
- Same results with closed-form and numerical integration

Impact of closed-form risk



- The approximation error decreases when increasing the number of parameters of numerical integration
- But the cost of num. int. increases nearly linearly
- The closed-form is always much faster

Conclusion and future work

- Adapted an unsupervised approach for linear classifier training that *minimizes the classifier risk* to NLP
- Derived a closed-form risk estimator
- Shown that proper weights initialization is required
- Validated the weakly supervised method on 2 NLP tasks

Conclusion and future work

- Completely unsupervised: combine gradient descent with particle swarm optimization
- Alternative to autoencoders and RBMs for unsupervised training of hidden layers in deep networks
- Speed-up: approx. “light-speed” GMM training + approx. gradient propagation in GMMs
- Remove Gaussianity assumption by using N Gaussians per class
- Derive closed-form for multi-class risk

Thank you for your attention !

#synalp_nancy
lina.rojas@eng.cam.ac.uk
cerisara@loria.fr