Low-Rank Regularization for Sparse Conjunctive Feature Spaces: An Application to Named Entity Classification

A. Primadhanty\textsuperscript{1} \quad X. Carreras\textsuperscript{2} \quad A. Quattoni\textsuperscript{2}

\textsuperscript{1}Universitat Politècnica de Catalunya
\textsuperscript{2}Xerox Research Centre Europe
Challenge

Conjunction of sparse elementary features
down
very sparse

Example: Named Entity Classification

A shipload of 12 tonnes of rice arrives in [Umm Qasr port] in the Gulf

\[
\phi_1(l) \quad \phi_e(e) \quad \phi_r(r)
\]

down
sparse
sparse
sparse
Approaches

\[ \ell_1 \text{ or } \ell_2 \]

unseen conjunctions?
Contributions

Low-rank regularization for sparse conjunctive feature spaces

Propagate weight to unseen conjunctions

Learning algorithm

Convex relaxation of the low-rank minimization function

Experiments

Improvement over $\ell_1$ & $\ell_2$
Task

**Given:**
\[ x = (l, e, r) \]

**Goal:**
Classify \( x \) into one entity class \( y \) in the set \( \mathcal{Y} \)

A shipload of 12 tonnes of rice arrives in [Umm Qasr port] in the Gulf

\[ l \quad e \quad r \]

\[ \downarrow \]

\[ y? \]
Classifier

Log-Linear Model

\[ \Pr(y \mid x; \theta) = \frac{\exp\{s_\theta(x, y)\}}{\sum_{y'} \exp\{s_\theta(x, y')\}} \]

\( s_\theta : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R} \) is scoring function of entity tuples with a candidate class

\( \theta \) are parameters of this function
Scoring Function

Feature-based linear model

\[ s_\theta(x, y) = \phi(x) \cdot w_y \]

\( \phi : X \rightarrow \{0, 1\}^n \) is a feature function representing entity tuples in an \( n \)-dimensional binary feature space

\( \theta = \{w_y\}_{y \in Y} \) are weight vector for each class
Scoring Function

Left-right context model

$$s_\theta(\langle l, e, r \rangle, y) = \phi_l(l)^T W_y \phi_r(r)$$

$\phi_l \in \mathbb{R}^{d_1}$ is a feature function representing left contexts

$\phi_r \in \mathbb{R}^{d_2}$ is a feature function representing right contexts

$W_y \in \mathbb{R}^{d_1 \times d_2}$ is weight matrix for each class, such that $\theta = \{W_y\}_{y \in \mathcal{Y}}$
Low Rank Parameter Matrices

\[ W_y = \begin{bmatrix} u_{11} & \cdots & u_{1k} \\ \vdots & \ddots & \vdots \\ u_{d_1} & \cdots & u_{d_1 k} \end{bmatrix} \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_k \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & \cdots & v_{1d_2} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ v_{k1} & \cdots & \cdots & v_{kd_2} \end{bmatrix} \]

Consider that \( W_y \) has rank \( k \)

\( U_y \in \mathbb{R}^{d_1 \times k} \) and \( V_y \in \mathbb{R}^{d_2 \times k} \) are orthonormal projections

\( \Sigma_y \in \mathbb{R}^{k \times k} \) is a diagonal matrix of singular values
Score Function - Rewritten

Left-right context model

\[ s_\theta(\langle l, e, r \rangle, y) = \phi_l(l)^\top W_y \phi_r(r) \]
Score Function - Rewritten

Left-right context model

\[ s_\theta(\langle l, e, r \rangle, y) = \phi_l(l)^\top W_y \phi_r(r) \]

\[
\begin{bmatrix}
  l_1 & \cdots & l_{d_1}
\end{bmatrix}
\begin{bmatrix}
  u_{11} & \cdots & u_{1k} \\
  \vdots & \ddots & \vdots \\
  u_{d_1} & \cdots & u_{d_1k}
\end{bmatrix}
\begin{bmatrix}
  \sigma_1 & \cdots & 0 \\
  \vdots & \ddots & \vdots \\
  0 & \cdots & \sigma_K
\end{bmatrix}
\begin{bmatrix}
  v_{11} & \cdots & v_{1d_2} \\
  \vdots & \ddots & \vdots \\
  v_{k1} & \cdots & v_{kd_2}
\end{bmatrix}
\begin{bmatrix}
  r_1 \\
  \vdots \\
  r_{d_2}
\end{bmatrix}
\]

Rank \( k \rightarrow \text{intrinsic dimensionality} \) of the inner product behind the score function
Adding Entity Features

One parameter matrix per feature tag and class label, i.e. \( \theta = \{W_{t,y}\}_{t \in T, y \in \mathcal{Y}} \)

\[
s_\theta(\langle l, e, r \rangle, y) = \sum_{t \in \phi_e(e)} \phi_1(l)^\top W_{t,y} \phi_r(r)
\]

Parameters: tensor

Rank defined by matricization
Learning The Parameters

**Objective Function**

\[
\arg\min_W L(W) + \tau R(W)
\]

\(L(W)\) is a convex **loss function** (negative log-likelihood)

\(R(W)\) is a **regularizer**

\(\tau\) is a constant that trades off error and capacity

Minimizing rank \(\rightarrow\) non-convex function

\[\downarrow\]

**nuclear norm**: convex relaxation

(Srebro & Shraibman, 2005)
Experimental Settings

**Task**
- Named Entity Classification

**Data**
- Annotated English CoNLL

**Training**
- Minimal supervision (seeds) + large unlabeled data
| Class  | 10-30 Seed |
|--------|------------|
| PER    | clinton, dole, arafat, yeltsin, wasim akram, lebed, dutroux, waqar younis, mushtaq ahmed, croft |
| LOC    | u.s., england, germany, britain, australia, france, spain, pakistan, italy, china |
| ORG    | reuters, u.n., oakland, puk, osce, cincinnati, eu, nato, ajax, honda |
| MISC   | russian, german, british, french, dutch, english, israeli, european, iraqi, australian |
| O      | year, percent, thursday, government, police, results, tuesday, soccer, president, monday, friday, people, minister, sunday, division, week, time, state, market, years, officials, group, company, saturday, match, at, world, home, august, standings |

For each entity class, the seed of entities for the **10-30** set.
### Experimental Settings

| Task       | Named Entity Classification |
|------------|----------------------------|
| Data       | Annotated English CoNLL    |
| Training   | Minimal supervision (seeds) + large unlabeled data |
| Evaluation | Mentions of unseen entities |
CoNLL 2003 English Corpus

Most entities in each set are non-ambiguous.

*Entities : unique candidate entities
Almost all seen entities that appear in dev can be directly classified as the same class.

*Entities: unique candidate entities
AVG-F1 on dev set using different seed set for training, comparing $\ell_1$, $\ell_2$ and nuclear-norm (NN) regularizer. Feature set: elementary features and all conjunctions of entity tags and left-right contexts (cluster & PoS), window size = 1

Seed set: number of examples per entity class (and $3 \times$ of non-entity examples)
Results on dev set

Only full conjunctions of left-right contexts (cluster), window size = 1

Elementary features and all conjunctions of entity tags and left-right contexts (cluster), window size = 1

Elementary features and all conjunctions of entity tags and left-right contexts (cluster & PoS), window size = 1

Only full conjunctions of entity tags and left-right contexts (cluster), window size = 1

Elementary features and all conjunctions of entity tags and left-right contexts (cluster), window size = 2

Elementary features and all conjunctions of entity tags and left-right contexts (cluster & PoS), window size = 2
Results on test set

F1 performance on test set using “all” seed set for training, with best setting (based on results on dev) for each regularizers.
Avg. F1 on development for increasing dimensions, using the best low-rank model in development set trained with all seeds.
Feature conjunctions in dev set

- conjunctions in dev
Feature conjunctions in dev set

- Red: conjunctions in dev that are unseen in train (with 10 seeds)
- Yellow: conjunctions in dev that are seen in train (with 10 seeds)
Feature conjunctions in dev set

- conjunctions in dev that are **unseen** in train (with 10 seeds) and has **zero weight**
- conjunctions in dev that are **seen** in train (with 10 seeds)
- conjunctions in dev that are **unseen** in train but assigned **non-zero weight** by model trained on 10 seeds
Feature conjunctions in dev set

- Conjunctions in dev that are **unseen** in train (with 10 seeds) and has **zero weight**
- Conjunctions in dev that are **seen** in train (with 10 seeds)
- Conjunctions in dev that are **unseen** in train but assigned **non-zero weight** by model trained on 10 seeds
Conclusion

Low-rank regularization framework for sparse conjunctive feature spaces

Tensors
Nuclear-norm

Experimented on learning entity classifiers

Compare to $\ell_1$ and $\ell_2$ penalties $\rightarrow$ better results
Illustrated weight propagation to unseen conjunctions

Future works: explore different tensor transformations
Thank you!