Pointwise HSIC: A Linear-Time Kernelized Co-occurrence Norm for Sparse Linguistic Expressions [EMNLP2018]

Sho Yokoi (Tohoku U./RIKEN AIP)*, Sosuke Kobayashi (PFN), Kenji Fukumizu (ISM), Jun Suzuki *, Kentaro Inui *

yokoi@ecei.tohoku.ac.jp

1809.00800 https://bit.ly/2SfpZmv github.com/cl-tohoku/phsic pip install phsic-cli

Computing Co-occurrence Strength in NLP

General Settings

Observed: paired data $\mathcal{D} = \{(x_i, y_i)\}$ $\sim \mathbf{P}_{XY}$

x_1	y_1
x_2	y_2
⋮	
x_n	y_n

Task: compute co-occurrence strength of a pair (x, y)

x	y	?

Collocation Extraction [Manning&Schütze,'99]

word bigrams from corpora

Have	you
you	ever
⋮	
York	?

which pairs are collocation?

New	York	✓
in	the	

Dialogue Response Selection [Lowe+,'15]

input-response message pairs from SNS

I'm hungry!	Let's have lunch
Will it rain today?	It's about to rain
⋮	
I love this manga	I don't know

which response is the best?

input by users	response candidates	
I've lost my wallet	I saw it at the …	✓
I've lost my wallet	I'm so sleepy	
I've lost my wallet	I don't know	


De facto Measure: Pointwise Mutual Information

Mutual Information

$$\mathrm{MI}(X,Y) = \mathrm{KL}[\mathbf{P}_{XY} \| \mathbf{P}_X \mathbf{P}_Y] = \mathop{\mathbf{E}}_{(x,y)}\left[\log \frac{\mathbf{P}_{XY}(x,y)}{\mathbf{P}_X(x)\mathbf{P}_Y(y)}\right]$$

Pointwise Mutual Information

$$\mathrm{PMI}(x,y;X,Y) = \log \frac{\mathbf{P}_{XY}(x,y)}{\mathbf{P}_X(x)\mathbf{P}_Y(y)}$$

co-occurrence **actual**

co-occurrence **by chance**

$$\widehat{\mathrm{PMI}}_{\mathrm{MLE}}(x,y) = \log \frac{n \cdot \#(x,y)}{\#(x,\cdot)\#(\cdot,y)}$$

just **counting**

- ✓ **easy to learn**
- ✗ inapplicable to sparse expressions

$$\widehat{\mathrm{PMI}}_{\mathrm{RNN}}(x,y) = \log \frac{\widehat{\mathbf{P}}_{\mathrm{RNN}}(y|x)}{\widehat{\mathbf{P}}_{\mathrm{RNN}}(y)}$$

using **RNNs** [Li+,'16]

- ✗ tough to learn
- ✓ **applicable to sparse expressions**

Proposed Measure : Pointwise HSIC

HSIC [Gretton+,'05]

$$\mathrm{HSIC}(X,Y;k,\ell) = \mathrm{MMD}^2_{k,\ell}[\mathbf{P}_{XY}, \mathbf{P}_X\mathbf{P}_Y]$$
$$= \mathop{\mathbf{E}}_{(x,y)}\left[(\phi(x)-m_X)^\top C_{XY}(\psi(y)-m_Y)\right]$$
$$= \mathop{\mathbf{E}}_{(x,y)}\left[\mathop{\mathbf{E}}_{(x',y')}[\tilde{k}(x,x')\tilde{\ell}(y,y')]\right]$$

$\phi(x) := k(x,\cdot)$, $\psi(y) := \ell(y,\cdot)$

$m_X := \mathop{\mathbf{E}}_{x}[\phi(x)]$, $m_Y := \mathop{\mathbf{E}}_{y}[\psi(y)]$

$C_{XY} := \mathop{\mathbf{E}}_{(x,y)}\left[(\phi(x)-m_X)(\psi(y)-m_Y)^\top\right]$

Kernels
$k\colon \mathcal{X}\times\mathcal{X}\to\mathbb{R}$
$\ell\colon \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$

Pointwise HSIC

$$\mathrm{PHSIC}(x,y;X,Y,k,\ell) = (\phi(x)-m_X)^\top C_{XY}(\psi(y)-m_Y)$$
$$= \mathop{\mathbf{E}}_{(x',y')}[\tilde{k}(x,x')\tilde{\ell}(y,y')]$$

Kernels centered on data

$$\tilde{k}(x,x') := k(x,x') - \mathop{\mathbf{E}}_{x}[(x,x')] - \mathop{\mathbf{E}}_{x'}[(x,x')] + \mathop{\mathbf{E}}_{x,x'}[(x,x')]$$

Dependence of (X,Y)

the difference between $\mathbf{P}_{XY}$ and $\mathbf{P}_X\mathbf{P}_Y$

Co-occurrence of (x,y)

the contribution of (x,y) to the dependence of (X,Y)

Requires Very Short Learning Time

estimators are reduced to a simple matrix calculation

when using cosine similarity
→ can estimate in RKHS directly

$$\widehat{\mathrm{PHSIC}}_{\mathrm{RKHS}}(x,y;k,\ell) = (\phi(x)-\overline{\phi(x)})^\top \widehat{C}_{XY}(\psi(y)-\overline{\psi(y)})$$

when using non-linear kernels
→ use incomplete Cholesky decomposition

$$\widehat{\mathrm{PHSIC}}_{\mathrm{ICD}}(x,y;k,\ell) = (\boldsymbol{a}-\overline{\boldsymbol{a}})^\top \widehat{C}_{\mathrm{ICD}}(\boldsymbol{b}-\overline{\boldsymbol{b}})$$

Kernels are Available

- **Cosine Similarity** between **Sentence Vectors**
  - Sentence vectors [Kiros+,'15; Dai&Le,'15; Iyyer+,'15; Hill+,'16; Cer+,'18]
  - Sum of word vectors [Mikolov+,'13; Pennington+,'14; Bojanowski+,'17]
  - Many pre-trained models are off-the-shelf!
- **Structure Kernels**
  - [Collins&Duffy,'02; Bunescu&Mooney,'06; Moschitti,'06]
- **Combinations**

Applicable to Sparse Expressions

PHSIC smooths the matching using kernels

	add scores	deduct scores
$\widehat{\mathrm{PMI}}(x,y;\mathcal{D}) = \log \frac{n\cdot\sum_i \mathbb{I}[x=x_i \wedge y=y_i]}{\sum_i \mathbb{I}[x=x_i]\sum_i \mathbb{I}[y=y_i]}$	$\mathcal{D} = \{\ldots, (x_i, y_i), \ldots\}$	$\{\ldots, (x_i, y_i), \ldots, (x, y), \ldots\}$
$\widehat{\mathrm{PHSIC}}(x,y;\mathcal{D},k,\ell) = \frac{1}{n}\sum_i \widehat{\widetilde{k}}(x,x_i)\widehat{\widetilde{\ell}}(y,y_i)$	$\{\ldots, (x_i, y_i), \ldots, (x, y), \ldots\}$	$\{\ldots, (x_i, y_i), \ldots, (x, y), \ldots\}$

exact match

smooths the matching **by kernels**

Experiments

1. Dialogue Response Selection [Lowe+,'15]

$n = 5\times10^5$

	Learning Time [s]	Predictive Performance [R@2]
PMI (w/RNN)	13054.9	0.52
PHSIC (cos of USE)	1.8	0.57

✓ thousands of times faster! ✓ while outperforming PMI

$n = 10^3$

	Learning Time [s]	Predictive Performance [R@2]
PMI (w/RNN)	49.0	0.21
PHSIC (cos of USE)	0.0	0.56

✓ much more robust to data sparsity

Kernels in Experiments

- **Cosine Similarity** between Sum of **Word Vectors**
  $k(x,x') = \cos\left(\sum_{w\in x}\boldsymbol{v}(w), \sum_{w\in x'}\boldsymbol{v}(w)\right)$
- **Gaussian Kernel** between Sum of **Word Vectors**
  $k(x,x') = \exp\left(-\left\|\sum_{w\in x}\boldsymbol{v}(w) - \sum_{w\in x'}\boldsymbol{v}(w)\right\|^2 / 2\sigma^2\right)$
- **Cosine Similarity** between **Sentence Vectors**
  $k(x,x') = \cos\left(\boldsymbol{v}(x), \boldsymbol{v}(x')\right)$

fastText [Bojanowski+,'17; Grave+,'18]

Universal Sentence Encoder [Cer+'18] w/**Deep Averaging Network** [Iyyer+,'15]

2. Parallel Corpus Filtering

BLEU score after filtering (3M ⟼ 1M)

random	39.82 (-1.20)
fast_align	40.56 (-0.46)
PHSIC (RBF kernel of fastText)	40.95 (-0.07)

baseline using **word alignment** [Dyer+,'13]

using all (3M) training data ▲41.02

✓ PHSIC reduces # of training data to 1/3, almost without sacrificing BLEU