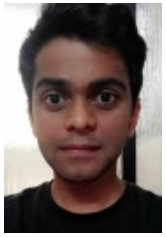
http://www.cs.princeton.edu/~arora/
Group website: unsupervised.cs.princeton.edu
Blog: www.offconvex.org
Twitter: @prfsanjeevarora

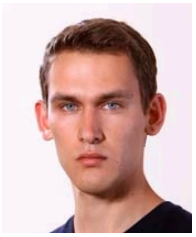
Support: NSF, ONR, Simons Foundation, Schmidt Foundation, Amazon Research, Mozilla Research. DARPA/SRC

Theory for representation learning

Sanjeev Arora

Princeton University and Institute for Advanced Study


Hrishi


Misha

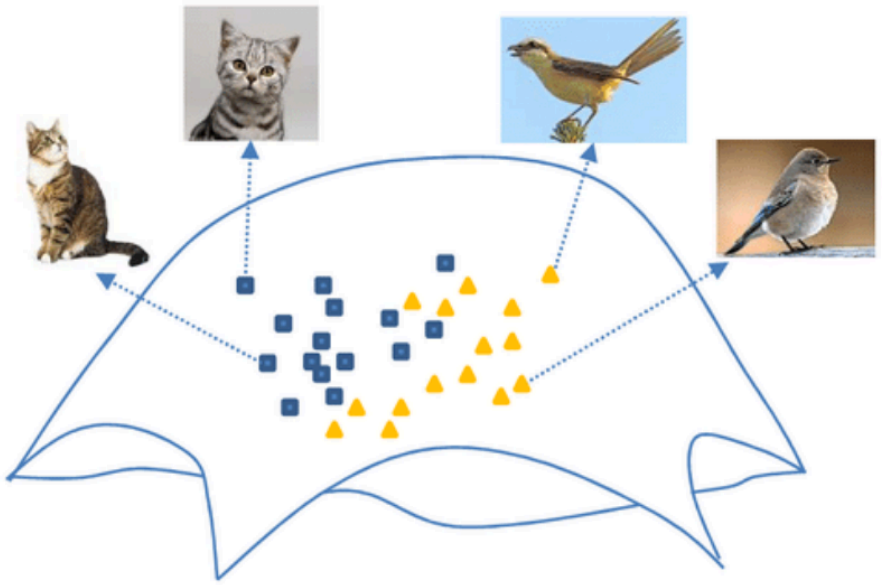

Orestis

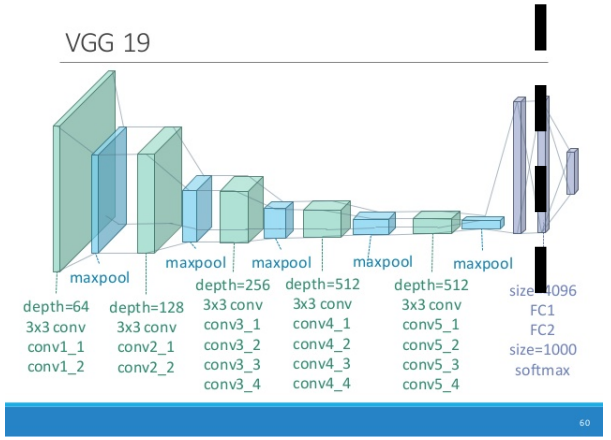

Nikunj

Paper 1: A theoretical analysis of contrastive unsupervised representation learning (CURL)" [A., Hrishikesh Khandeparkar, Mikhail Khodak (CMU), Orestis Plevrakis, Nikunj Saunshi ICML'19)

Paper 2: A graph-theoretic analysis of CURL. A, Plevrakis, Saunshi 2019 manuscript.

Big motivation for GANs/VAEs etc: Semantic Embeddings f: {images} ➔ embeddings, s.t. f(x) is good representation of x for classification tasks





Can we bypass generative models and learn semantic embeddings directly?

Preferably as good as those from "Headless Well-trained Net"

Conceptual hurdle:
Why does learning to do A help you do B later on?

Example:

A = Learn embeddings

B = Use them in new classification tasks

Surprisingly, this is hard to capture* for Machine Learning Theory

(*except if you go hardcore, full Bayesian, but even then many conceptual difficulties, eg bits of precision)

Conceptual difficulties with generative model approach (or related ones, eg info. theory)

Evidence they don't actually learn the distribution, but suppose they did, sort of…

$p_\theta(x|h)$ x = image; h = seed = "semantic embedding" of x

$p_\theta(h|x)$ Way to generate semantic embedding of x

[A., Risteski, blogpost 2017]: If want linear classification on h to work with accuracy ϵ then must learn $p_\theta(h|x)$ with accuracy ϵ^2

(follows from Pinsker's Inequality)

Contrastive Unsupervised Representation Learning (CURL)

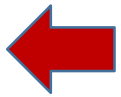
QuickThoughts [Logeswaran & Lee, ICLR'18] "like word2vec.." "Self-supervised"

Using text corpus train deep representation function f to minimize

$$\mathbb{E}\left[\log\left(1+e^{f(x)^T f(x^-)-f(x)^T f(x^+)}\right)\right]$$

x, x^+ are adjacent sentences, x^- is random sentence from corpus

("High inner product for adjacent sentences; low inner product for random pairs of sentences.")



Similar ideas work for embedding molecules, genes, social nets

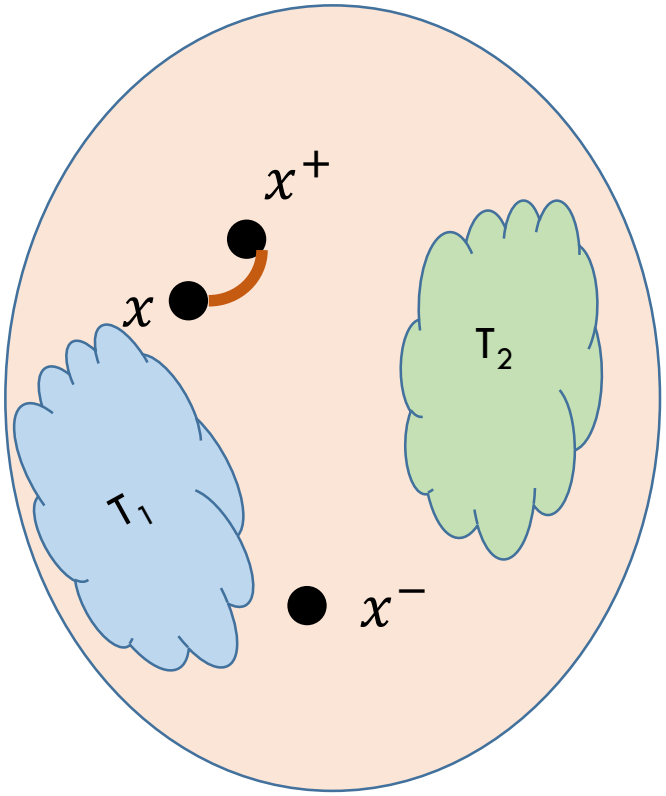
[For image embeddings, Wang-Gupta'15 use video…]

Learns representations by leveraging contrast between ”similar” and “dissimilar” (eg, random) pairs of datapoints.

Graph-Based Framework for Understanding CURL

“Why do learnt representations help in downstream classification tasks?”

Doing Task A later helps in Task B??



Graph G= (V, E)

V = all possible datapoints

(eg, sentences with < 30 words).

E = "similar" pairs.

Nature's sampling process:

Repeat M times.

Reveals e = (x, x^+) from some distribution on E.

Reveal node x^- from some distribution on V

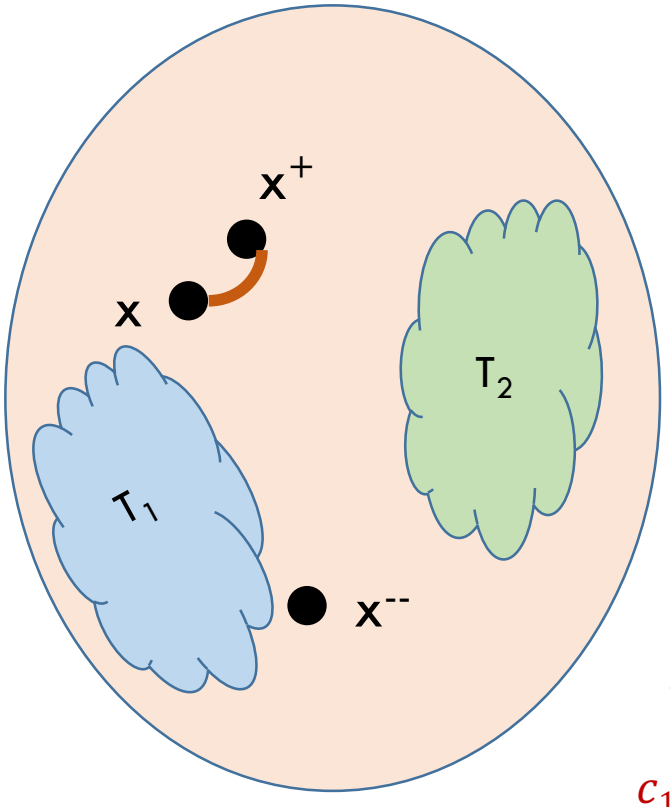
Task A: Run CURL on the M samples

Task B:

Nature picks datapoints from two classes T_1, T_2, represents each via f, and trains logistic classifier to separate them.

CURL may not have seen any data from T_1, T_2

Conceptual Framework



Graph G= (V, E)
V = all possible datapoints
E = "similar" pairs.

Nature's sampling process:
Repeat M times.
Reveals e = (x, x^+) from some distribution on E.
Reveal node x^- from some distribution on V

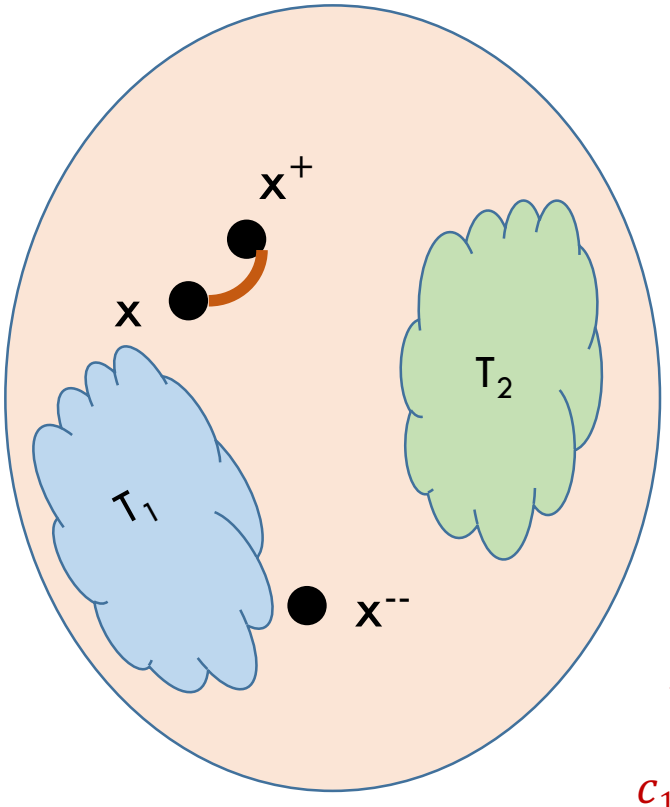
$\rho(c)$ = prob. of picking class c

$c_1, c_2 \sim \rho$, $x, x^+ \sim D_{c_1}(x)$, $x^- \sim D_{c_2}(\mathrm{x})$

Test time:

Nature picks datapoints from two classes T_1, T_2 and asks algo. to learn to classify using logistic classifier.

Reminiscent of Multiview/cotraining setup



Graph G= (V, E)
V = all possible datapoints
E = "similar" pairs.

Nature's sampling process:
Repeat M times.
Reveals e = (x, x^+) from some distribution on E.
Reveal node x^- from some distribution on V

$\rho(c)$ = prob. of picking class c

$c_1, c_2 \sim \rho$, $x, x^+ \sim D_{c_1}(x)$, $x^- \sim D_{c_2}(\mathrm{x})$

Unpacking a little…

"Similarity" ≈ "Tend to go together (or not) for random class."
(will later relax this)

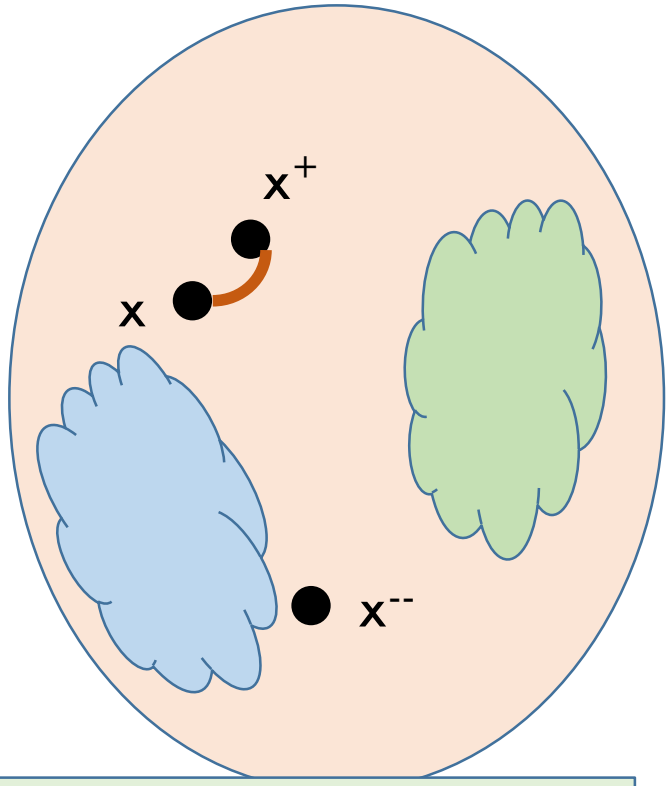
The analysis…

Part 1: Why CURL makes sense even though graph is humongous, even infinite.

Part 2: Why CURL representations can solve the classification tasks

(NB: Will ignore computational cost, and just analyse quality of representations that have low training loss..)

Analysis (a): Why CURL makes sense even though G is humongous



Graph G= (V, E)
V = all possible datapoints
E = "similar" pairs.

Nature's sampling process:
Repeat M times.

Reveals e = (x, x^+) from some distribution on E.
Reveal node x^- from some distribution on V

$c_1, c_2 \sim \rho$, $x, x^+ \sim D_{c_1}(x)$, $x^- \sim D_{c_2}(x)$

$\rho(c)$ = prob. of picking class c

$$L_{un}(f) = \mathop{\mathbb{E}}_{\substack{(x,x^+)\sim D_{sim} \\ x^- \sim D_{neg}}} \left[\log\left(1 + e^{f(x)^T f(x^-) - f(x)^T f(x^+)}\right)\right]$$

Thm: If M > d R(F)/ε
Then L_{un}(f) on samples tracks that on the full graph within ε
(R( ) = Rademacher Complexity)

Analysis (b): Relating classification accuracy to low value of $L_{un}(f)$

$$L_{un}(f) = \mathop{\mathbb{E}}_{\substack{(x,x^+)\sim D_{sim} \\ x^-\sim D_{neg}}} \left[\log\left(1 + e^{f(x)^T f(x^-) - f(x)^T f(x^+)}\right)\right]$$

Theorem: Average Binary Task Guarantee

With probability at least $1-\delta$, for all $f \in \mathcal{F}$

$$L_{sup}^{\mu}(\hat{f}) \leq \frac{1}{1-\tau}\left[L_{un}(f) - \tau + \varepsilon\right]$$

$\tau =$ collision probability for pair of random classes (usually small)

Translation: Every f with low unsup. Loss gives low classification loss on avg binary task c_1, c_2 using a logistic classifier

(Note: Precision requirements more benign than in generative models.).

(Reminder) Logistic classifier on binary task. *

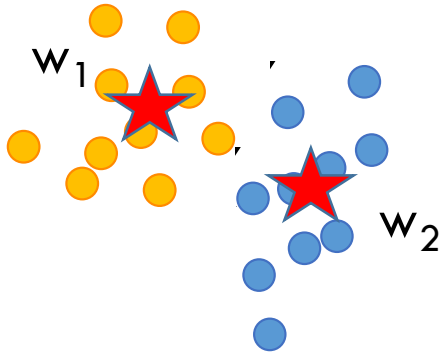
Given: Data labeled with 0/1

Trains vectors w_1, w_2.

Output on input x is the following:

$$P(y=1) = \frac{e^{\langle w_1, x\rangle}}{e^{\langle w_1, x\rangle} + e^{\langle w_2, x\rangle}}$$

$$P(y=2) = \frac{e^{\langle w_2, x\rangle}}{e^{\langle w_1, x\rangle} + e^{\langle w_2, x\rangle}}$$



* Aka "softmax," usually used as the top layer of deep nets

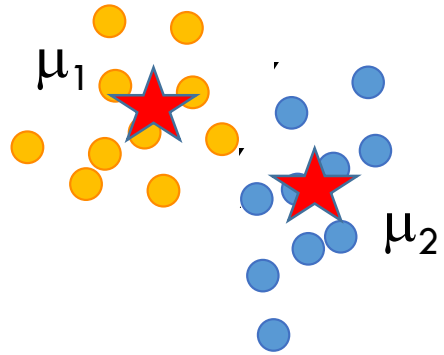
Theorem: Average Binary Task Guarantee

With probability at least $1-\delta$, for all $f \in \mathcal{F}$

$$L_{sup}^{\mu}(\hat{f}) \leq \frac{1}{1-\tau}[L_{un}(f) - \tau + \varepsilon]$$

Pf idea 1: mean classifiers for 2-way classifications

Instead of training best w_1, w_2 to minimize logistic loss, set w_i =mean of representation of samples from c_i



$$\mu_c = \mathop{\mathbb{E}}_{x \sim \mathcal{D}_c} f(x)$$

$$L_{sup}^{\mu}(task, f) = \mathop{\mathbb{E}}_{(x,c)\sim task} \log(1 + \sum_{c' \neq c} e^{f(x)^T(\mu_{c'} - \mu_c)})$$

$$L_{sup}^{\mu}(f) = \mathop{\mathbb{E}}_{task} L_{sup}^{\mu}(task, f)$$

Theorem: Average Binary Task Guarantee

With probability at least $1-\delta$, for all $f \in \mathcal{F}$

$$L_{sup}^{\mu}(\hat{f}) \leq \frac{1}{1-\tau}\left[L_{un}(f) - \tau + \mathcal{E}\right]$$

Pf Idea 2 Key step: Jensen's inequality ($\varphi(\mathbb{E}[X]) \leq \mathbb{E}[\varphi(X)]$.)

$$\log\left(1 + e^{f(x)^T \mu_{c-} - f(x)^T \mu_{c+}}\right) \leq \mathop{\mathbb{E}}_{\substack{x^+ \sim \mathcal{D}_{c+} \\ x^- \sim \mathcal{D}_{c-}}} \log\left(1 + e^{f(x)^T f(x^-) - f(x)^T f(x^+)}\right)$$

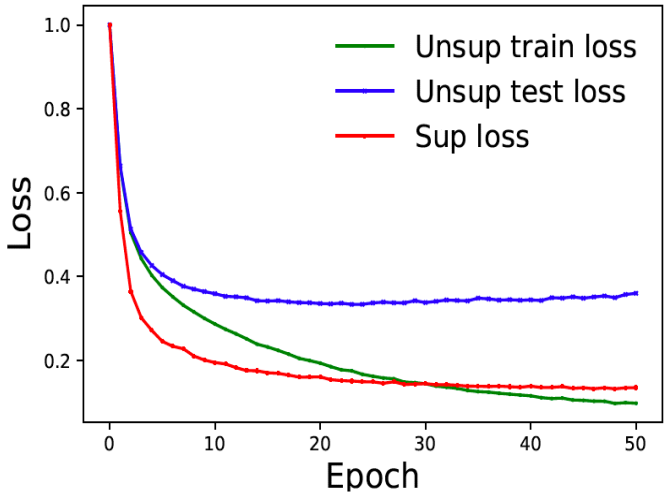
Sup loss of mean classifier

Unsup loss

NB: # of labeled samples needed is sample complexity of linear classification (can be made precise; see paper)

Experiments/Test of Theory

- $\mathcal{F}$ = GRU, VGG-16
- Controlled setting, where distributional assumptions hold.
- WIKI-3029: classes are the articles datapoints are sentences.
- CIFAR 100



Representations trained on the full multiclass problem, using labeled data

		Supervised			Unsupervised		
		Tr	μ	μ-5	Tr	μ	μ-5
Wiki-3029	AVG-2	97.8	97.7	97.0	97.3	97.7	96.9
	AVG-10	89.1	87.2	83.1	88.4	87.4	83.5
	TOP-10	67.4	59.0	48.2	64.7	59.0	45.8
	TOP-1	43.2	33.2	21.7	38.7	30.4	17.0
CIFAR-100	AVG-2	97.2	95.9	95.8	93.2	92.0	90.6
	AVG-5	92.7	89.8	89.4	80.9	79.4	75.7
	TOP-5	88.9	83.5	82.5	70.4	65.6	59.0
	TOP-1	72.1	69.9	67.3	36.9	31.8	25.0

Quick-Thoughts

Dataset	Method	b=2	b=5	b=10
IMDB	CURL	89.2	89.6	89.7

Blocks can help "in the wild"

New paper (to be released soon)..

Key weakness so far: x, x^+ are indep. samples from the same class.
(i.e., "similarity" ≡ "tend to end up on same side at test time")

New assumption: Class can consist of subclasses; x, x^+ are indep. samples from a subclass.
(So "similarity" ≡ tend to co-occur in subclasses)

CURL ⇒ classification harder to establish;
uses SVM duality and spectral graph theory.

Conclusions

- A first cut theory for formalization of representation learning; minimalistic assumptions!
- Future work: Extensions to more intricate settings (eg lattice structure or metric structure among classes)?
- Extensions to other "Task A vs Task B" settings? Transfer learning/meta learning/cycle GANs/.. Etc.

Resources: articles on www.offconvex.org

Grad lec. notes on theory of deep learning fall'17 and fall'18