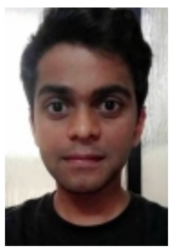
[MIT IDSS Colloquium 2019]

Paper: A theoretical analysis of contrastive unsupervised representation learning"
[A., Hrishikesh Khandeparkar, Mikhail Khodak (CMU), Orestis Plevrakis, Nikunj Saunshi ARXIV'2019)

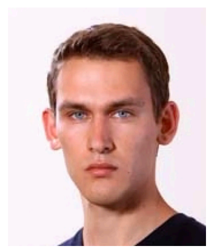
Theory for representation learning

Sanjeev Arora

Princeton University and Institute for Advanced Study



Hrishi



Misha

Orestis

Nikunj

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 Resarch, Mozilla Research. DARPA/SRC

Talks I am not giving today…

“What is Machine Learning?”/ “The math of Machine Learning and Deep Learning.”

“Toward theoretical understanding of deep learning…”

(various youtube versions, including 2-hr ICML’18 tutorial; lots of new work on understanding of deep learning landscape, generalization, optimization etc.)

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

Example:

A = major in math

B = earn $$$ on Wall St.

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

(*except if you go hardcore, full Bayesian.
But, even then unclear how to interpret the phenomena
we'll see in next few slides.)

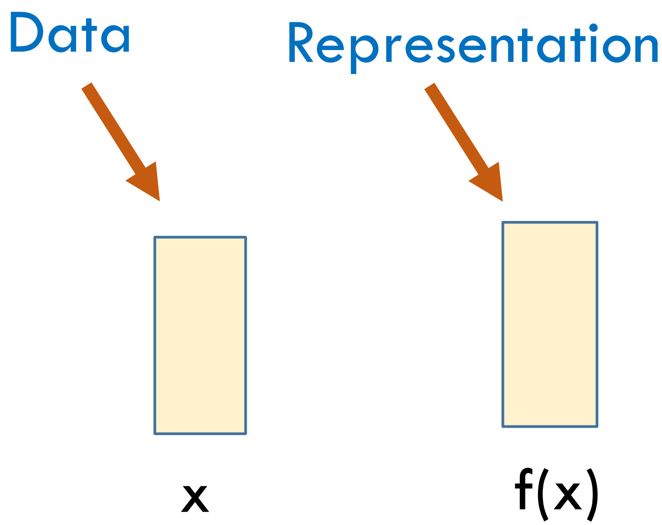
Talk Overview

- Part 1: Representation learning. Desired goals and need for new theoretical framework
- Part 2: The Lore of representations/embeddings (empirical results from vision, NLP).
- Part 3: Our new framework; minimalistic yet surprisingly powerful.
- Part 4: Some experiments

PART 1: REPRESENTATION LEARNING AND ITS GOALS (AND NEED FOR NEW THEORETICAL FRAMEWORK…)

("Solving Task A later helpful in doing Task B. ")

Data Representation



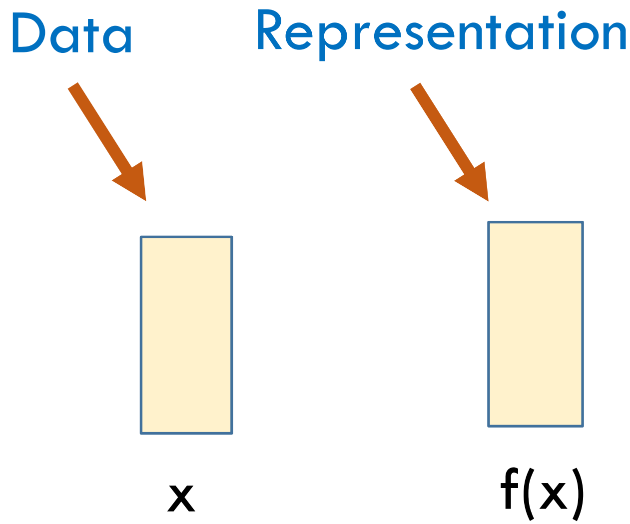
Solve *other* *tasks using these representations.*

(f = representation function)

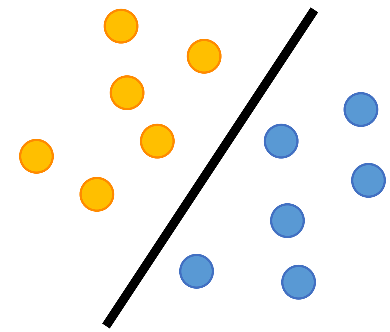
Often called "embeddings" (e.g., text embedding)

(e.g., image → Pattern of visual cortex activations it leads to…)

Powerful Data Representation



(f = representation function)

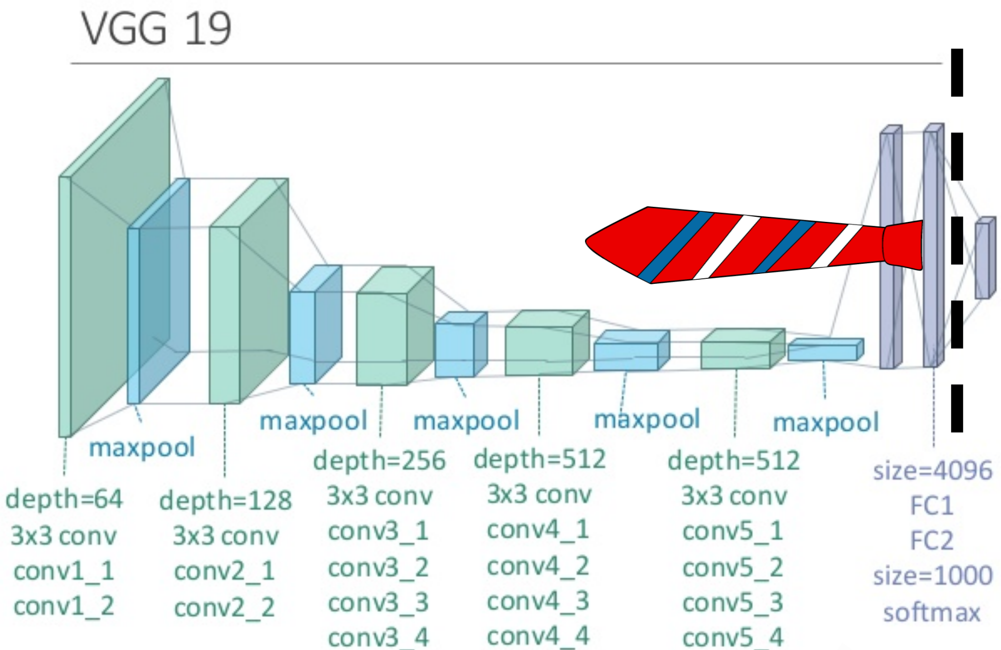


Allows new classification tasks to be solved well using linear classifier on f(x).

method to bring semantic content "to the surface".

Classic example: Kernel SVMs. "Lift" data to kernel representation, classify using linear classifier.

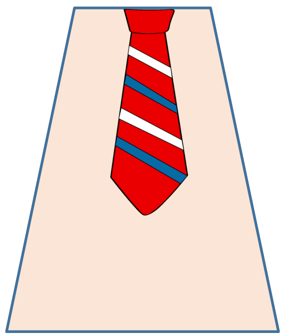
With lots of labeled data, deep nets learn powerful representations



Trained on labeled dataset ImageNet: (10^3 classes, 10^3 examples ea.)

- Classification accuracy abysmal if trained with 2 classes (⇒ other 998 classes important for learning the right representation!)
- Vector on penultimate layer useful: solves new unrelated tasks via linear classifier!

With lots of labeled data, deep nets learn powerful representations



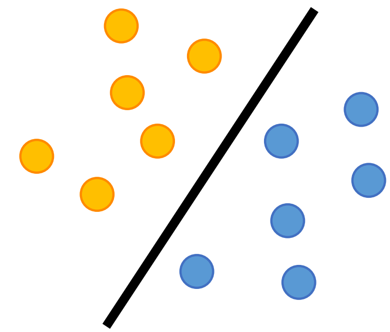
"Headless well-trained deep net" (Gold Standard representation for rest of the talk)

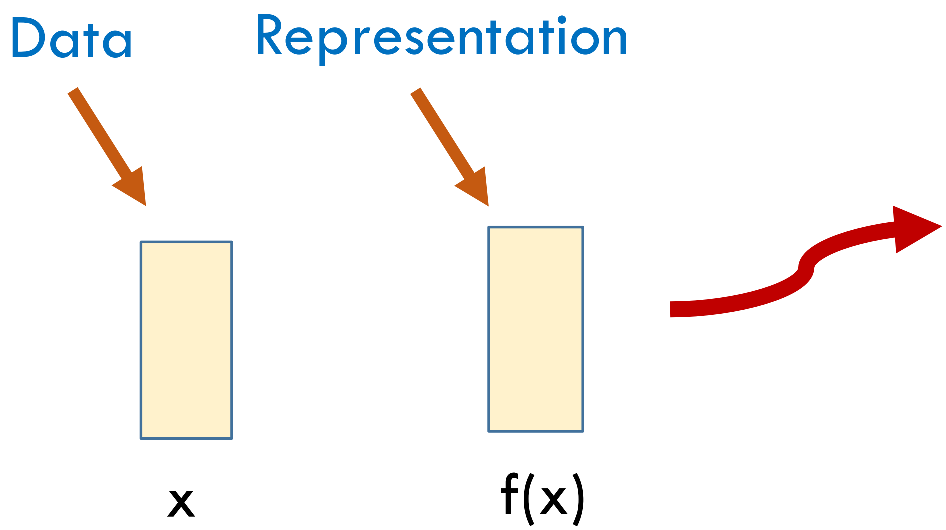
Can we use only unlabeled data to learn equally good representations?

Trained on labeled dataset ImageNet: (10^3 classes, 10^3 examples ea.)

- Classification accuracy abysmal if trained with 2 classes (⇒ other 998 classes important for learning the right representation!)
- Vector on penultimate layer useful: solves new unrelated tasks via linear classifier!

Powerful Data Representation



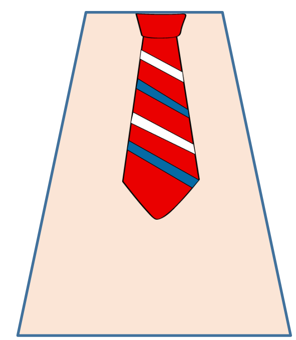


(f = representation function)

Allows new classification tasks to be solved well using linear classifier on f(x).

Ideally, as good as headless well-trained deep net!

What theory can predict such a thing?



Standard ML paradigm

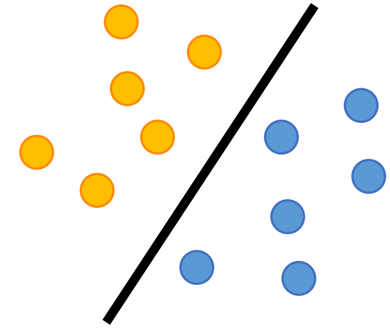
Train learner on some data; test on held-out data.

Training/test involve the same objective, and involve i.i.d. samples from same distribution

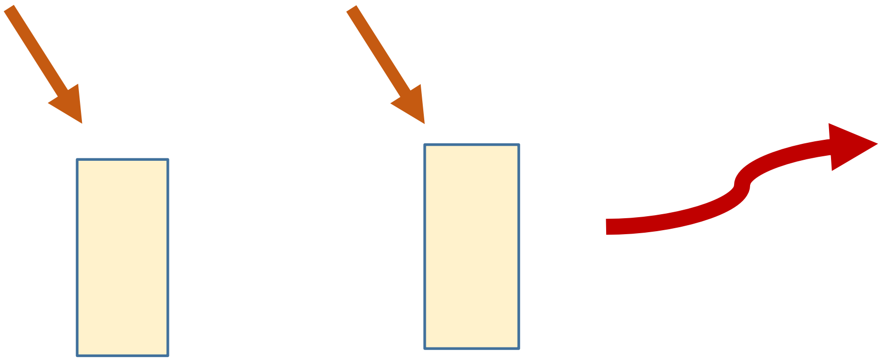
Training loss − Test loss = Generalization error

For fixed model, RHS → 0 as # training samples → ∞

Powerful Data Representation



Data Representation

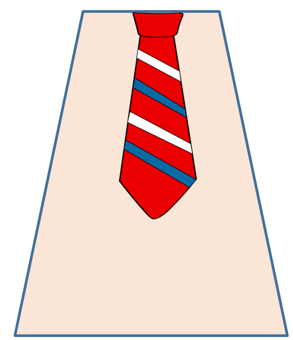


x f(x)

(f = representation function)

Allows new classification tasks to be solved well using linear classifier on $f(x)$.

Ideally, as good as
headless
well-trained deep net!



What theory can predict this?

** test and train involve different objectives…*

** don't know test tasks while training…*

What theory can predict this?
** test and train involve different objectives…*
** don't know test tasks while training…*

→ Much theory on semi-supervised methods: training uses unlabeled data as well as labeled data from downstream task (e.g., kernel learning)

Also popular: Generative models (e.g. topic models, language models, VAE, etc.)

- Training and test objective are same: log (Pr[Data]), or "perplexity"
- Unclear why this objective should suffice for representation learning in practice; see discussion by A. + Risteski on offconvex.org.
- Above methods (eg QuickThought) do not appear to do Bayesian reasoning

PART 2: THE LORE OF SEMANTIC EMBEDDINGS…

(Created via solving Task A, helpful in doing Task B.
Not much theoretical analysis…)

Ex1: Word embeddings via language models

Idea: Using large corpus (eg, Wikipedia), train a model to predict part of text from adjacent text.

Example: *"I went to a café and ordered a…. "*

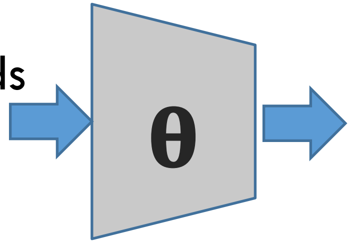
(In learning to do this, model implicitly picks up on grammar rules, common sense etc.)

Ex1: Word embeddings via language models

Baby word2vec [Mikolov et al'13]

"I went to a café and ordered a…. "

Preceding words
$w_1, w_2, .., w_5$



Distribution on all English words

$$\Pr[w|w_1 \ldots w_5] \propto \exp(\frac{1}{5}\sum_i \langle v_w, v_{w_i}\rangle)$$

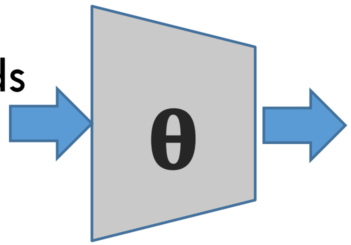
(implicit normalizing term; will ignore for simplicity)

$\theta = \{ v_w \in R^{300}$: **w an English word**$\}$
("word embeddings")

Loss $\ell(\theta)$: Reciprocal of Probability assigned by model to Wikipedia = $w_1 w_2 w_3 \ldots w_N$

Baby word2vec (contd)

Preceding words
$w_1, w_2, .., w_5$



Distribution on all English words

$$\Pr[w|w_1 \ldots w_5] \propto \exp\left(\frac{1}{5}\sum_i \langle v_w, v_{w_i}\rangle\right)$$

$\boldsymbol{\theta}$ = { $v_w \in R^{300}$: **w an English word**}
("semantic vectors")

(for simplicity, ignore normalizing term..)

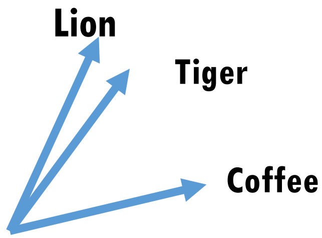
Loss $\ell(\boldsymbol{\theta})$: Reciprocal of Probability assigned by model to Wikipedia = $w_1 w_2 w_3 \ldots w_N$

$$\prod_{i=6}^{N} \Pr[w_i | w_{i-5}, \ldots, w_{i-1}]$$

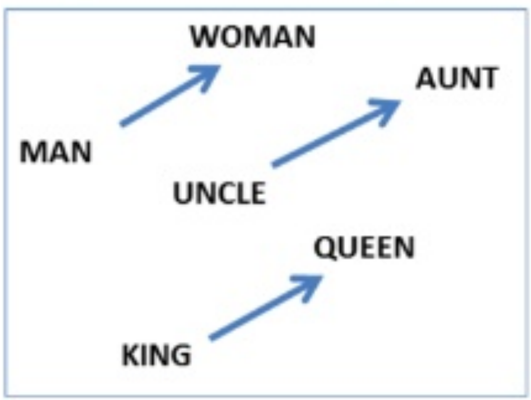
$$= \exp\left(\sum_{i=6}^{N} \sum_{j=1}^{5} \frac{1}{5} \langle v_{w_i}, v_{w_{i-j}} \rangle\right)$$

Training method: negative sampling. Tries to give high inner product to word pairs occurring nearby, and low inner product to random pairs of words.

Magical properties of word embeddings



Cosine of angle captures human estimates of "similarity"[Deerwester et al'90]



Word analogies can be solved via Linear algebra on word embeddings

Man : woman :: King : ??

Word vector space for different languages (e.g., English, French) can be meaningfully aligned via linear transformation [Lample et al'18, Arttextxe et al'18]

Ex 2: Bizarre method for image embeddings

"Unsupervised representation learning by predicting image rotations"

[Gidaris et al, ICLR'18][Zhang et al. '19]

Idea: Train ConvNet on following task.

Input: Image And its rotation by either 90, 180, or 270 degrees

Desired Output: which of the three rotations was applied.

Train ConvNet on this "self-labeled" data. Representations learnt this way are quite powerful (compared to gold standard)!

Ex 3: Sentence embeddings via QuickThoughts

[Logeswaran & Lee, ICLR'18] "like word2vec.."

Using text corpus (eg Wikipedia) 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

("Make adjacent sentences have high inner product, while random pairs of sentences have low inner product.")

We call such methods
"Contrastive Learning"
(word2vec-like)
State of the art!

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

Sentence embeddings capture human notions of similarity

1) The tiger rules this jungle.

2) Milk flowed out from the bottle.

3) Carnegie was a generous man.

4) A lion hunts in a forest.

5) Pittsburgh has great restaurants, doe

Note: No words in common!

Similarity scores via inner product of embeddings

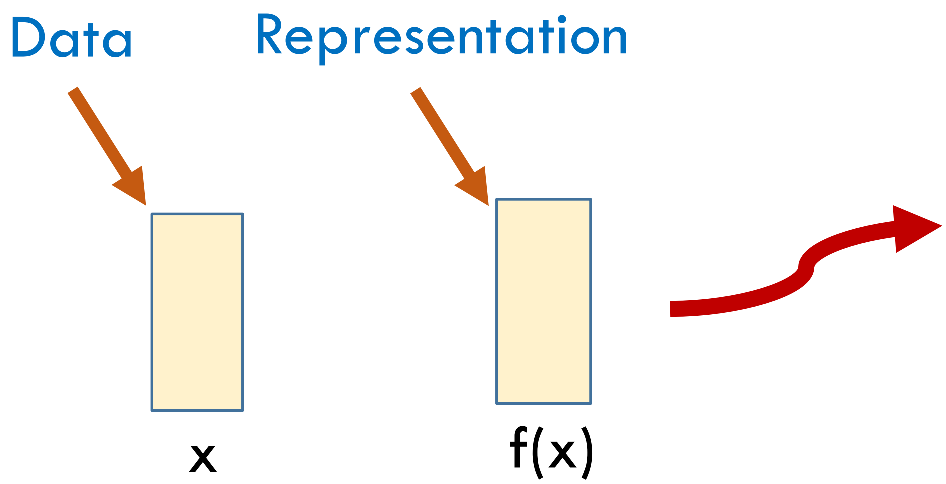
See articles on offconvex.org for more on embeddings…

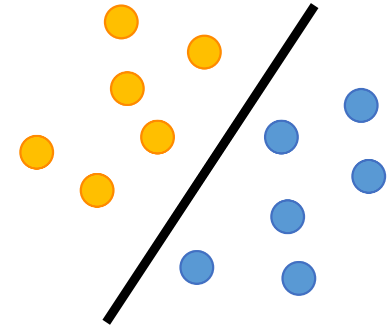
(Again, training objective seems unrelated to test objective (which is in human heads)…)

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

PART 3: NEW FRAMEWORK FOR CONTRASTIVE UNSUPERVISED LEARNING : THE PARTS

Goal: Powerful Data Representations





Allows new classification tasks to be solved well using linear classifier on f(x).

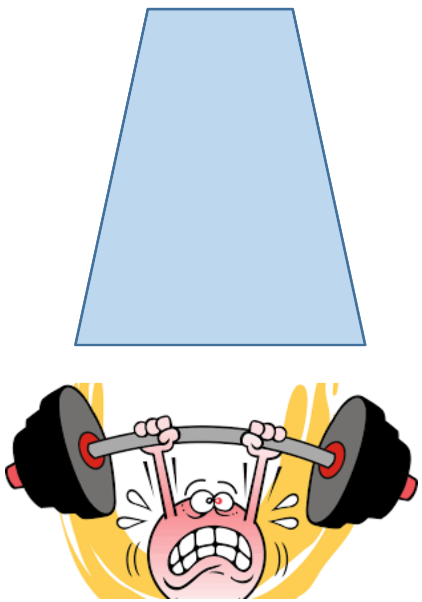
Part 1 of theory: Available data consists of:
Pairs (x, x+) of "similar" inputs.
Random pairs of inputs (x, x-) treated as "dissimilar."

"Contrastive Data."

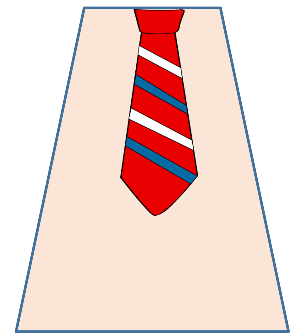
Part 2: Learn best representation from function class $\mathcal{F}$

Available data: Pairs (x, x+) of "similar" inputs.
Random pairs of inputs (x, x-) treated as "dissimilar."

Fix particular deep net architecture (eg., ResNet 50 of certain size)

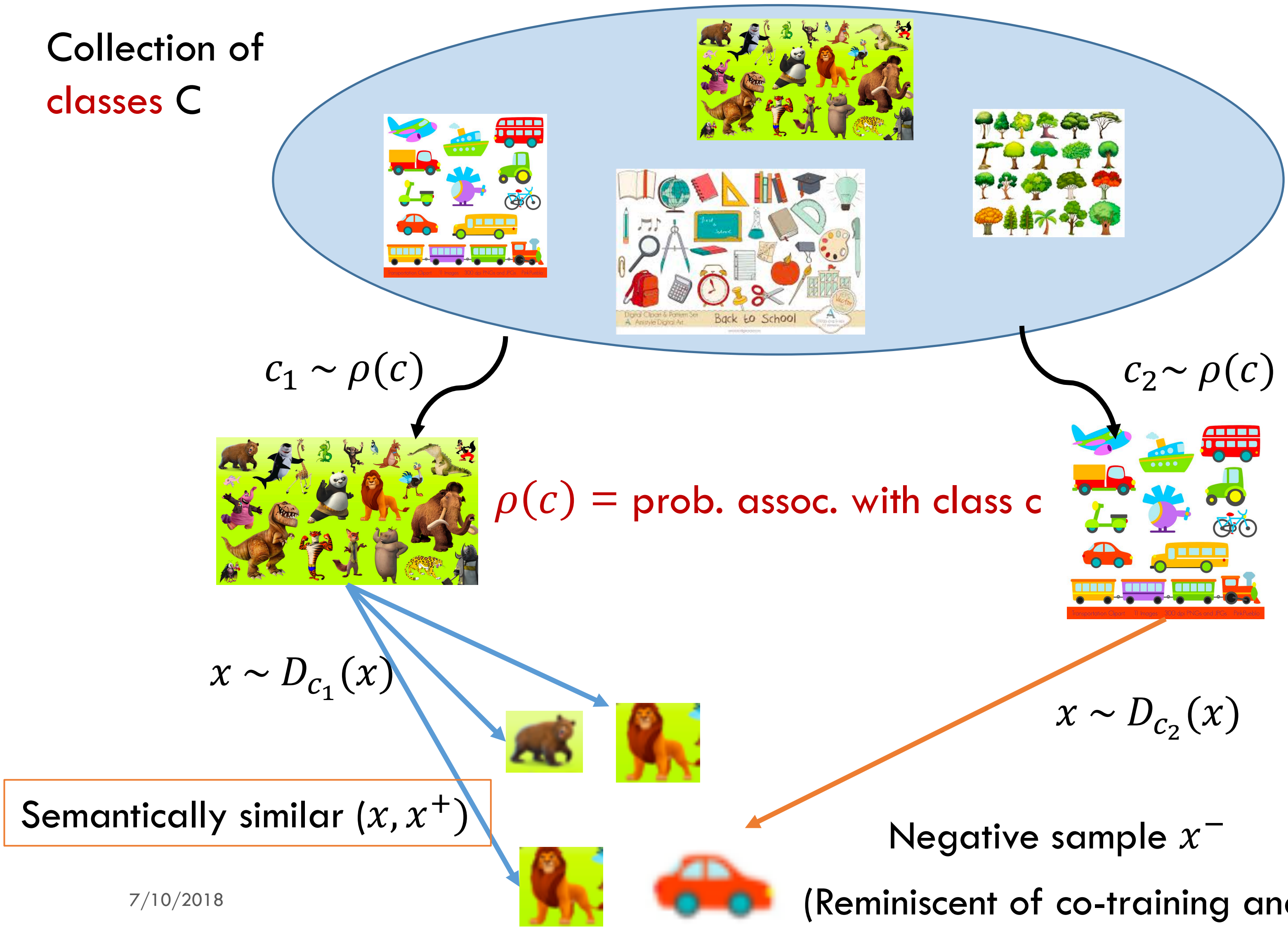


Want: Method to use available data to train this architecture to produce representation f.



Goal: Understand why minimizing unsupervised loss helps for supervised task

Part 3.1: Assumption about "Semantically similar" pairs



- Classes sampled from C according to $\rho(c)$
- Samples x from from c according to $x \sim D_c(x)$

(Reminiscent of co-training and Multiview assumptions…)

Part 3.1: Assumption about “Semantically similar” pairs

- A ‘class’ defines distrib. D_c on datapoints;

$D_c(x)$ = Prob. of seeing datapoint x in c (note: x may lie in many classes, which can overlap arbitrarily)

Key assumptions

- “Similar pairs”: Pick c_1 according to ρ and then two indep. samples x, x^+ from c_1 according to $D_{c_1}(x)$
- “Negative Sample/Dissimilar pairs” : Pick c_2 according to ρ and then x^- according to $D_{c_2}(x)$

Part 3.2) What downstream classification tasks are of interest?

(For now, restrict to 2-way classification)

- Nature picks random pair of distinct classes $(c_1, c_2) \propto \rho(c_1)\rho(c_2)$
- Pick k_1 i.i.d. samples from $D_{c_1}()$, and k_2 iid samples from D_{c_2}, where k_1/k_2 can depend on pair (c_1, c_2).

Part 3.3) Evaluation of representation: Pick random binary task as above. Solve by training logistic classifier on the representations.

(Theory extends to hinge loss...)

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

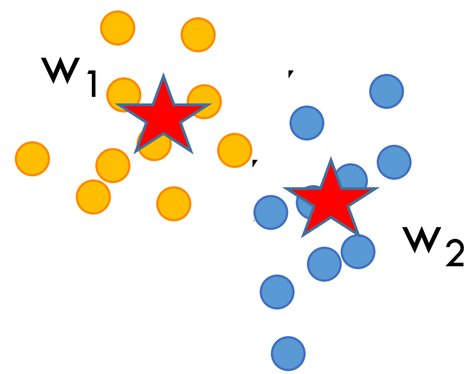
Aside: Logistic classifier on binary task. *

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

Part 3.4) Our method to learn representation (like QuickThought)

Unsupervised Loss:

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

Main Qs: How does best f do in classification tasks?

Empirical Objective (for M sam

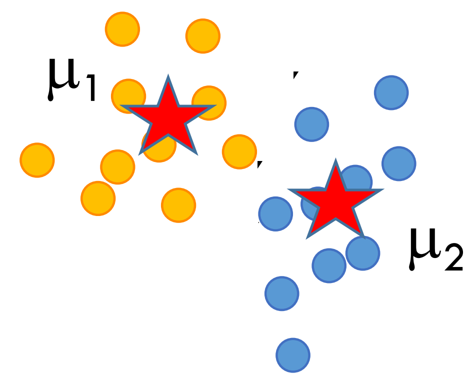
$$\widehat{L}_{un}(f) = \frac{1}{M}\sum_{i=1}^{M}\left[\log\left(1 + e^{f(x_i)^T f(x_i^-) - f(x_i)^T f(x_i^+)}\right)\right]$$

Notes 1) Unlabeled data is cheap! Assume M large enough that the above two optima are approx. same once we fix a class of f's (eg ResNet50 of certain size). Exact M computable using rademacher complexity…
2) We ignore computational cost of minimizing $\widehat{L}_{un}$

Mean classifiers for 2-way classifications

(in practice is almost as good as optimum classifier, and much nicer to analyse…)

When solving classification, Instead of training w_1, w_2 to minimize logistic loss, just set w_i to be the mean representation of samples from c_i



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

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

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

Warmup: Simple result

Useful since unsup. loss is low in many settings..

$$L_{sup}^{\mu}(f) \leq \frac{1}{1-\tau}(L_{un}(f) - \tau), \ \forall f \in \mathcal{F}$$

"If unsupervised loss low, then avg. loss on classification tasks is low"

τ = collision probability for pair of random classes (usually small)

Key step: Jensen's inequality

$$\underbrace{\log\left(1 + e^{f(x)^T \mu_{c-} - f(x)^T \mu_{c+}}\right)}_{\text{Sup loss of mean classifier}} \leq \mathop{\mathbb{E}}_{\substack{x^+ \sim \mathcal{D}_{c+} \\ x^- \sim \mathcal{D}_{c-}}} \underbrace{\log\left(1 + e^{f(x)^T f(x^-) - f(x)^T f(x^+)}\right)}_{\text{Unsup loss}}$$

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

Handling case when $L_{un}()$ is not small.

$$L_{sup}^{\mu}(\widehat{f}) \leq L_{un}^{\neq}(f) + \frac{2\tau}{1-\tau} s(f) + \frac{1}{1-\tau} Gen_M \qquad \forall f$$

Term for $c^+ \neq c^-$

$s(f)$ is a notion of geometric variance among representations within classes

Type equation here.

Let $\Sigma(f,c)$ be the covariance matrix of $f(x)$ when $x \sim \mathcal{D}_c$ and

$$s(f) = \mathop{\mathbb{E}}_{c\sim\rho}\left[\sqrt{\|\Sigma(f,c)\|_2} \mathop{\mathbb{E}}_{x\sim\mathcal{D}_c} \|f(x)\|_2\right]$$

Guarantee is strong if we have

- Contrastive f
- Small collision probability
- Concentrated f
- More unlabeled data

(Empirically, we find representations are concentrated, so above bound can be stronger)

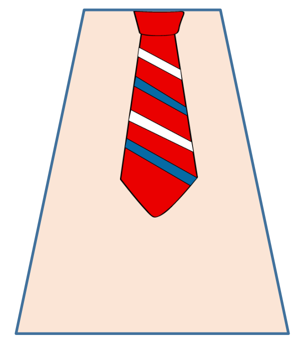
Dream result for analysis?

If $\widehat{f} \in \arg\min_{f\in\mathcal{F}} \widehat{L}_{un}(f)$ "Learnt representation"

then would like "Competitive with BEST representation"

$$L_{sup}(\widehat{f}) \leq \alpha L_{sup}(f) + \gamma\, Gen_M \qquad \forall f$$

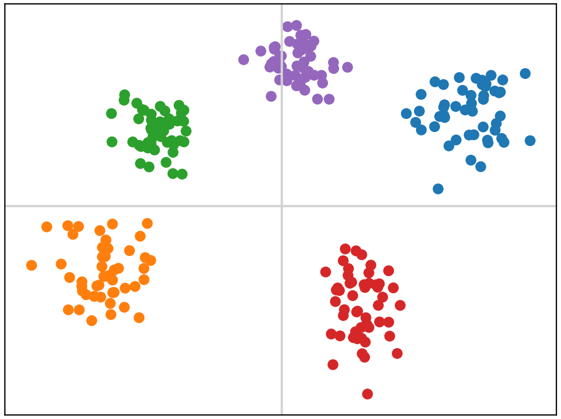
(2^{nd} term ➔ 0 as unlabeled data is cheap.
Unsup. representation would compete with best representation function f in the same class of circuits/deep nets)



Easy Thm: This is impossible for an arbitrary class of functions and arbitrary tasks…

Theory needs to work around this..

Progress toward dream result (under stronger assumption)



We can compete against headless well-trained deep net that produces representations "tightly concentrated" within classes and have high margin using mean classifier.).

Thm: for some f, σ^2 sub-gaussian in each class + low $(1+\widetilde{\Omega}\,(\sigma R))$-margin loss using mean classifier

=> low 1-margin loss for our representations.

(R : max norm of representations)

Extensions (briefly)

- Extends to k-way classification. Corresponding unsup. learning uses one similar pair and k-1 negative samples.
- A new unsup. objective based upon blocks of r similar datapoints. Allows a tighter bound.

Some experiments

Wiki-3029 database: Classes = 3029 articles on Wikipedia. Datapoints in a class = 200 sentences.

Only 5 labeled samples per class!

Train sentence representations; use to solve 2-way and 10-way classification tasks.

		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

(Similar experiments for CIFAR100 Image dataset, though supervised/unsupervised gap is larger)

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

Improving state of art text embeddings (QuickThought) via “block objective"

IMDB: 50k movie reviews.

QuickThought[Logeswaran-Lee’18] : learns representations using contrastive learning. Predicts IMDB ratings from review text via linear classification.

CURL: Our version of contrastive learning with blocks (treat each review as a block of similar sentences).

IMDB	CURL	89.2	89.6	89.7
	QT	86.5	87.7	86.7

Both models use same LSTM architecture.

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)?
- More empirical and theoretical development? Transfer learning/meta learning etc.?

Thank you!

Resources www.offconvex.org

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

Sample complexity benefit

$$\widehat{f} \in \arg\min_{f \in \mathcal{F}} \widehat{L}_{un}(f)$$

$$L^{\mu}_{sup}(\widehat{f}) \leq \frac{1}{1-\tau}\left(L_{un}(f) - \tau\right) + \boxed{\frac{1}{1-\tau} Gen_M}, \ \forall f \in \mathcal{F}$$

Gen_M is at most O(dR) * Supervised_Complexity(F) / M

(R : max norm of representations)

Significantly reduces labeled data requirement

Price of unlabeled data

Inherent issue because of lack of labels: Negative sample can be from the **same class** as similar pairs.

$$L_{un}(f) = (1-\tau)L^{\neq}_{un}(f) + \tau L^{=}_{un}(f)$$

Term for $c^+ \neq c^-$ Prob. of $c^+ = c^-$ Term for $c^+ = c^-$

To handle class collision, in addition to contrasting different classes, f must have "low variance" in each class

Handling class collision

$$L^{\mu}_{sup}(\widehat{f}) \leq L^{\neq}_{un}(f) + \frac{2\tau}{1-\tau} s(f) + \frac{1}{1-\tau} Gen_M$$

Where $s(f)$ is a notion of deviation of representations within classes

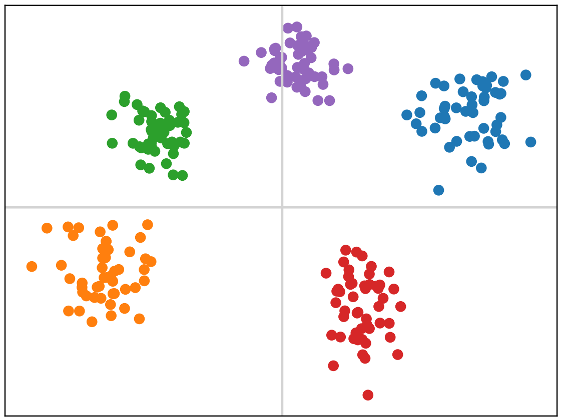
Let $\Sigma(f,c)$ be the covariance matrix of $f(x)$ when $x \sim \mathcal{D}_c$ and

$$s(f) = \mathop{\mathbb{E}}_{c \sim \rho}\left[\sqrt{\|\Sigma(f,c)\|_2} \mathop{\mathbb{E}}_{x \sim \mathcal{D}_c} \|f(x)\|_2\right]$$

Guarantee is strong if we have

- Contrastive f
- Small collision probability
- Concentrated f
- More unlabeled data

Progress toward dream result (under stronger assumption)



We can compete against f that has high margin with mean classifier and is highly concentrated in each class.).

Thm: σ^2 sub-gaussian in each class + low $(1 + \widetilde{\Omega}\,(\sigma R))$-margin loss for some f => low 1-margin loss for our representations.

(R : max norm of representations)