[ICTS Bangalore 2019]

Theory for representation learning

Sanjeev Arora

Princeton University and Institute for Advanced Study

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

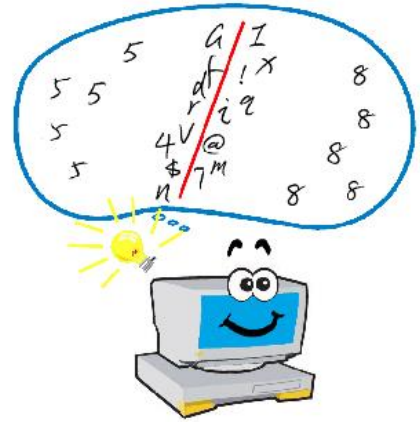
Support: NSF, ONR, Simons Foundation, Schmidt Foundation, Amazon Resarch, Mozilla Research. DARPA/SRC

Holy grail of ML: "High level" description of objects, using as little human supervision as possible

Hope: Allows ability to flexibly adapt to new tasks

Talk Overview

- Part 1: (Warmup) Lore of Word embeddings
- Part 2: Survey of representation learning and its goals (in vision, NLP) from a theory perspective.
- Part 3: New analysis framework; minimalistic yet surprisingly powerful.
- Part 4: Some experiments



The linear algebraic structure of word meanings [TACL'16]

Sanjeev Arora

Princeton University

Computer Science

Yuanzhi Li **Yingyu Liang** **Tengyu Ma** **Andrej Risteski**

(Funding: NSF and Simons Foundation)

What is meaning? How to test understanding?

- Solve analogies: man: woman :: king: ??

- Give more examples in the sequence:
  Japan Tokyo
  China Beijing
  Germany Berlin
  ….

Word embedding = representation of word's meaning as a vector.

Useful for these and many other tasks(machine translation, answering questions, image labeling etc.): successful example of unsupervised learning.

Test: Think of a word that co-occurs with:
Cow, drink, babies, calcium…

Distributional hypothesis of meaning, [Harris'54], [Firth'57]

Meaning of a word is determined by words it co-occurs with.

High dimensional word embedding for w:
$M_w(c)$ = Pr[words w, c co-occur in window of size 5 in corpus]

Better embeddings: "dimension reduced" version of above
(via SVD [Deerwester et al'90], neural nets, energy-based models,…)

Questions about word embeddings

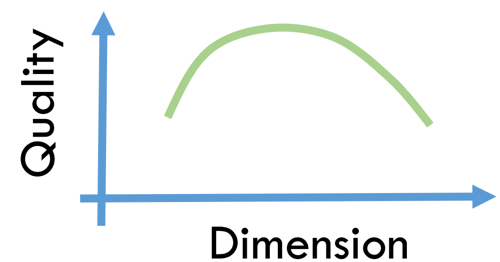
1. Why do they exist? (i.e., why can a 300-dimensional vector faithfully summarize distribution of 10^5 context words, giving efficient realization of Firth's idea?)

2. Why do semantic relations correspond to lines?

   ("queen" $\simeq$ "king" - "man" + "woman")

3. Why is there a sweet spot for dimension?

   Pointed out empirically already in [Dumais et al 1997]



This paper: "Explanation" via new Generative Model for Language.

Qs1: Why do low-dimensional word vectors exist?

[Church-Hanks'90] PMI(w, w') = log (P(w, w') / P(w)P(w'))

Embedding = 300-dim SVD

(GloVe, word2vec are fancier versions [Levy-Goldberg'14])

$$\begin{array}{cc} w & \left[\begin{array}{c} \\ \mathrm{PMI}(w,w') \\ \\ \end{array}\right] \\ & w' \end{array}$$

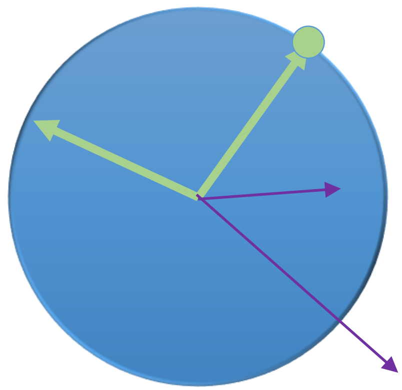
What property of language causes this 10^5 x 10^5 matrix to have approximate rank 300?

Main issue: Nonlinearity/logarithm

(If replace PMI(w, w') with Pr(w, w')
then explanation = "Topic Models".)

Generative model for language

(dynamic version of loglinear topic model, [Mnih-Hinton06])



"Semantic space" inside writer's head; each direction in R^d associated with a discourse (narrow "topic")

Each word w also associated with a vector v_w in R^d ("latent variable")

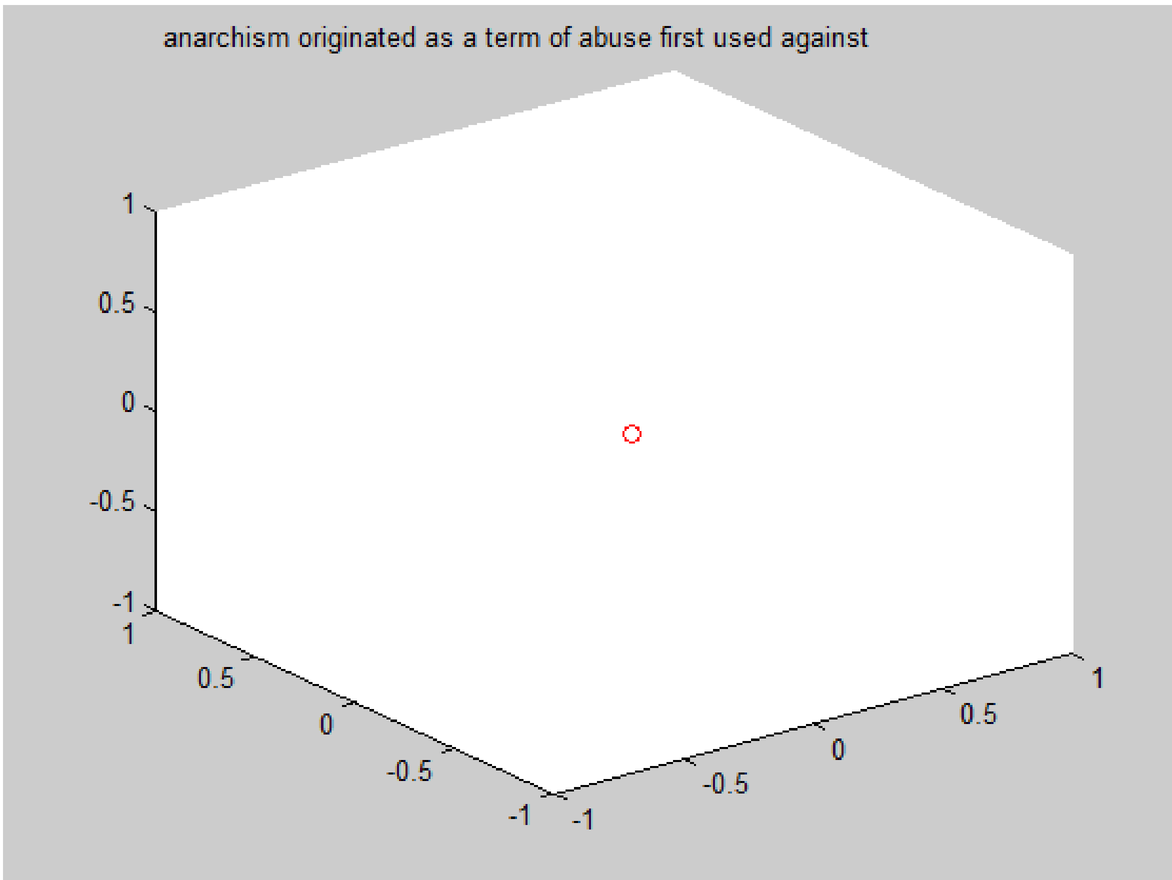
Corpus generated by a random walk of a discourse vector c on unit sphere in R^d $\Pr[w \text{ is output} \mid c_t] \propto \exp(v_w \cdot c_t)$

[Related: [Hashimoto, Alvarez-Melis, Jaakola TACL'16]

What distribution on bigrams does this process generate??

Generative model (illustration)

Discourse vector c_t does random walk;
if context vector at time t is c_t, $\Pr[w \text{ is output} \mid c_t] \propto \exp(v_w \cdot c_t)$



NB: Locally bag of words;

No syntax modeling.

Main Theorem

Assumptions

If discourse vector at time t is c_t, $\Pr[w \text{ is output}] \propto \exp(v_w \cdot c_t)$

(i) discourse space= unit sphere in R^d ; c_t doing slow random walk

(ii) v_w 's spatially isotropic

Empirically, fits with 5% error

Main Thm: $$\log(P[w,w']) = ||v_w + v_{w'}||^2/d - 2\log Z \pm \epsilon$$

$$\log(P[w]) = ||v_w||^2/d - \log Z \pm \epsilon$$

$$\boxed{PMI(w,w') = v_w \cdot v_{w'}/d \pm O(\epsilon)}$$

Fits with 17%

$\Rightarrow$ Norm of word vector determines f...
spatial orientation (which d...
determines "meaning".

See articles on offconvex.org for more on embeddings…

Part 2: Representation learning overview..

From now on will work with "minimalistic" assumptions; no explicit generative model for data…

Standard framework for ML

Training/test involve i.i.d. samples from same distribution

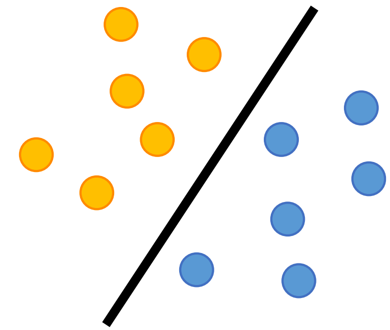
Training loss − Test loss = Generalization error

What if goal of learning is to be able to solve new tasks?

(Test and train involve different objectives…)

Examples: representation learning, transfer learning, meta learning..

This talk



New classification tasks using these representations

Data Representation

x f(x)

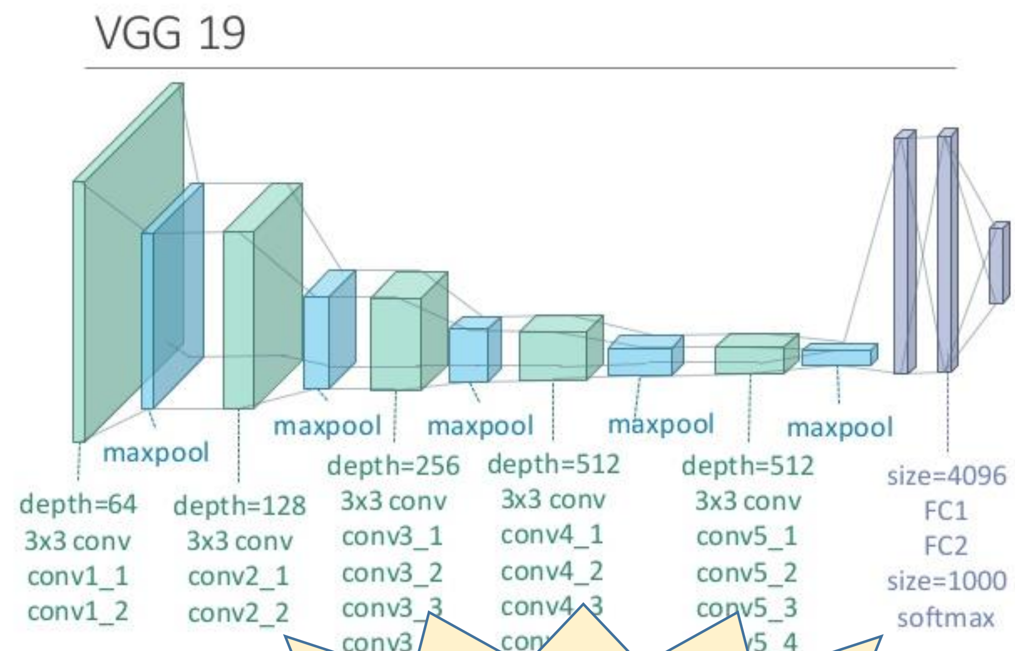
(f = representation function)

Powerful representation ⇒ allows new classif. tasks to be solved via linear classifier (with few labeled examples…)

Example: Kernel SVM

(Linear classification becomes possible after "lifting" x to kernel space)

Deep nets implicitly learn good representations



Can we learn such "gold-standard" representations with no labeled data?

Trained on ImageNet: (1000 classes, with 1000 examples each)

- Performance abysmal if trained with 2 classes
- Vector on penultimate layer (before softmax) a good representation in unrelated tasks!

How can we possibly learn a useful representation from unlabeled data, and without knowing downstream classification tasks?

→ Most theory work is on semi-supervised methods: training uses both labeled and unlabeled data. (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; see discussion by A. + Risteski on offconvex.org

Some interesting representation learning ideas that work well in practice…

"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.

Representations learnt by this "self-supervised" learning quite good compared to those learnt using supervised training (with labels)!

QuickThoughts [Logeswaran & Lee, ICLR'18]

(SoTA unsupervised sentence representation. "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"

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

Embeddings capture human notions of sentence 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 our head)...)

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

Rest of the talk based on

"A theoretical analysis of contrastive learning (unsupervised representation learning)"

[A., Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, Nikunj Saunshi 2019]

Hrishi

Misha

Orestis

Nikunj

The framework….

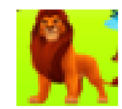
1) What are "semantically similar" pairs?

- World has collection of classes
  $\rho(c)$ = prob. assoc. with class c

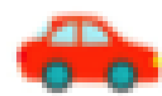
- Each class defines distrib. D_c on datapoints;

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

- "Similar pairs": Pick c according to ρ and then two indep. samples x, x' from c according to D_c



- "Negative samples" : Pick c according to ρ and then x from c according to D_c



(Reminiscent of co-training and Multiview assumptions…)

2) What downstream classification tasks are of interest?

(For now, restrict to 2-way classification)

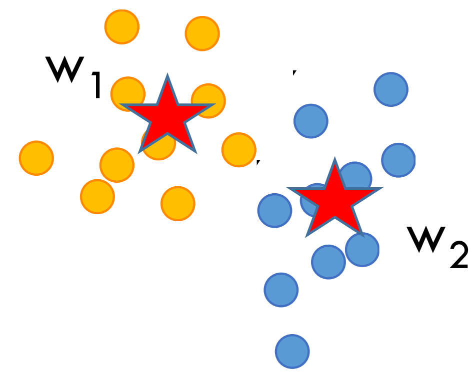
- Pick random pair of distinct classes (c_1, c_2) $\propto \rho(c_1)\rho(c_2)$
- Pick k_1 i.i.d. samples from $D_{c_1}()$, and k_2 iid samples from D_{c_2}, where k_1/k_2 can depend on pair (c_1, c_2).
- Test representations on this binary classification task.

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

(Theory extends to all usual convex losses…)

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

Aside: Logistic classifier on binary task
(Also, top layer of most deep nets)
Trains vectors w_1, w_2.
Output on input x is the following:



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

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

4) How to train your representation function

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 sa

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

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 $\widehat{L}_{un}$

Dream result for analysis?

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

then would like

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

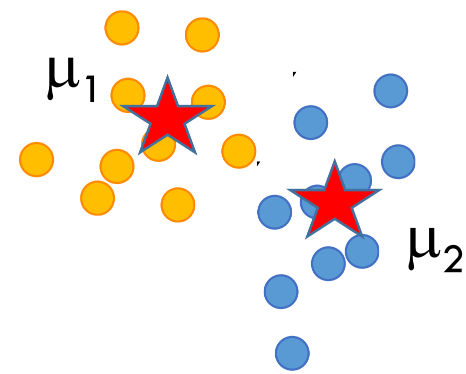
(2^{nd} term ➔ 0 since unlabeled data is cheap.
So our representation would compete with best representation function f in the same class of circuits/deep nets)

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

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..

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

"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

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

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.

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

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

Goes to 0 as # samples M rises.

$s(f)$ 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

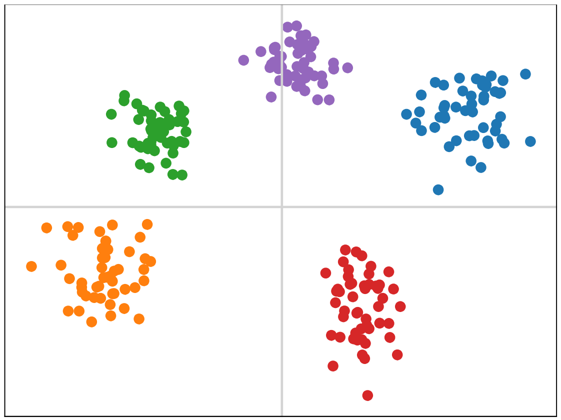
$$s(f) = \mathop{\mathbb{E}}_{c \sim \rho}\left[\sqrt{\|\Sigma(f,c)\|_2} \mathop{\mathbb{E}}_{x \sim \mathcal{D}_c} \|f(x)\|_2\right]$$

Guarantee is strong if we have

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

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

Progress toward dream result (under stronger assumption)



We can compete with gold-standard representations f that are "concentrated" within class and have high margin using mean classifier.).

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)

Extensions (briefly)

- Extends to all convex loss functions used in practice
- 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
	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

(Similar experiments for CIFAR100, 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
	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

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

IMDB: 50k movie reviews.

QuickThought[Logeswaran-Lee'18] : learns representations using contrastive learning. Predicts ratings (good/bad) via linear classification.

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

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

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

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

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

(R : max norm of representations)

Significantly reduces labeled data requirement

Price of using unlabeled data

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

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

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

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

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

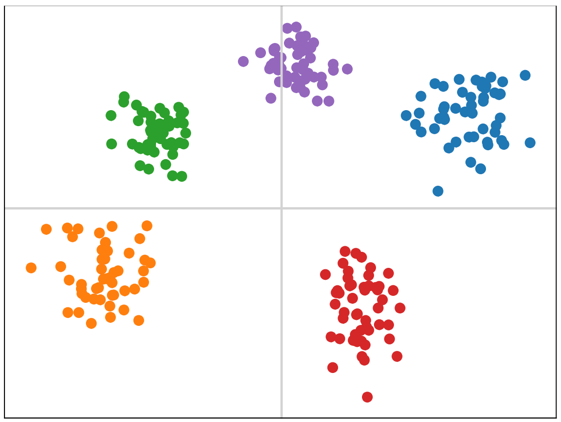
$$s(f) = \mathop{\mathbb{E}}_{c \sim \rho} \left[\sqrt{\|\Sigma(f,c)\|_2} \mathop{\mathbb{E}}_{x \sim \mathcal{D}_c} \|f(x)\|_2 \right]$$

Guarantee is strong if we have

- Contrastive f
- Small collision probability
- f vectors concentrated within classes
- More unlabeled data

Often holds in practice

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)