Performance guarantees for transferring representations

Daniel McNamara

PhD candidate, Australian National University and Data61
Fulbright Postgraduate Scholar visiting Carnegie Mellon University
Joint work with Nina Balcan, Carnegie Mellon University

March 31, 2017
Motivation

I want to automatically create separate photo albums of my dog Rufus and my cat Macy!

Maybe you can download one of those fancy neural network models from the Internet?

But there aren’t any photos of Rufus and Macy on the Internet! They’re unique!

C’mon, wouldn’t a model trained on all the images in the world ever have something in common with your photos?

Fine, I’ll try it. But I’m still going to make it special for Rufus and Macy! I guess we’ll find out if it works...
Motivation

- Want to learn a task for which labelled data is scarce, but have abundant data for another related task
- Transferring representations between tasks is empirically successful [DJV$^+$14, HGT$^+$14, GDDM14, BGL14]
- Natural language processing example: word embeddings outperform unigram features [QFZ$^+$15]
- Computer vision example: pre-trained neural network with fine tuning outperforms random initialization [YCBL14]
- When and why does this procedure work?
Notation

- $F$ is a class of representations, $f : X \rightarrow Z$ for $f \in F$
- $G$ is a class of specialized classifiers, $g : Z \rightarrow Y$ for $g \in G$
- $H := \{h : \exists f \in F, g \in G \text{ s.t. } h = g \circ f\}$, VC dimension $d_H$
- Source task $S$ and target task $T$ have labeling functions $h_S, h_T : X \rightarrow Y$ and input distributions $P_S, P_T$
- $m_S$ labelled points for $S$ and $m_T$ labelled points for $T$
- $R_S(h) := \mathbb{E}_{x \sim P_S}[h_S(x) \neq h(x)]$, $\hat{R}_S(h)$ is empirical risk on $S$
- $R_T(h) := \mathbb{E}_{x \sim P_S}[h_T(x) \neq h(x)]$, $\hat{R}_T(h)$ is empirical risk on $T$
Performance guarantees for transferring representations

Introduction

High-level idea

- Learn $\hat{f} : X \rightarrow Z$ from source task $S$
- Can we restrict the representation class $F$ when learning target task $T$?
- Use statistical learning theory to provide tighter risk upper bounds for $T$, inspired by [BDBCP07, Bax00, MPRP16]
Representation fixed by source task

- Learn $\hat{g}_S \circ \hat{f} \in H$ on $S$, extract $\hat{f} \in F$
- Then conduct empirical risk minimization over $G \circ \hat{f} := \{g \circ \hat{f} : g \in G\}$ on $T$, yield $\hat{g}_T := \arg\min_{g \in G} \hat{R}_T(g \circ \hat{f})$

**Theorem 1 (Risk upper bound for fixed representation)**

Let $\omega : \mathbb{R} \to \mathbb{R}$ be non-decreasing and $P_S, P_T, h_S, h_T, \hat{f}, G$ satisfy $\forall \hat{g}_S \in G$, $\min_{g \in G} R_T(g \circ \hat{f}) \leq \omega(R_S(\hat{g}_S \circ \hat{f}))$. Then with probability at least $1 - \delta$ over pairs of training sets for tasks $S$ and $T$,

$$R_T(\hat{g}_T \circ \hat{f}) \leq \omega(R_S(\hat{g}_S \circ \hat{f})) + 2\sqrt{\frac{2d_H \log(2em_S/d_H) + 2 \log(8/\delta)}{m_S}} + 4\sqrt{\frac{2d_G \log(2em_T/d_G) + 2 \log(8/\delta)}{m_T}}.$$

- If $\omega(R) = O(R)$, $R_S(\hat{g}_S \circ \hat{f})$ is a small constant, $m_S \gg m_T$ and $d_H \gg d_G$, bound in Theorem 1 is tighter than learning $T$ from scratch and using VC dimension-based risk bound
Neural network example with fixed representation

- Transfer lower-level weights learned on $S$, corresponding to $\hat{f}$
- Only the upper-level weights have to be learned on $T$
- Under network architecture and distributional assumptions, can define $\omega$ parameterized by constants $c$ and $\epsilon$
- $R_S(\hat{g}_S \circ \hat{f})$ reliably indicates usefulness of $\hat{f}$ if ‘defects’ of $\hat{f}$ cannot be hidden either through either low $P_S$ or low magnitude upper-level weights
Neural network example with fixed representation

- \( X = \mathbb{R}^n \) and \( Z = \mathbb{R}^k \), where \( 2k \leq n \)
- \( F \) is the function class s.t. \( f(x) = [a(w_1 \cdot x), \ldots, a(w_k \cdot x)] \), \( w_i \in \mathbb{R}^n \), \( a: \mathbb{R} \to \mathbb{R} \) odd, \( \hat{f}(x) := [a(\hat{w}_1 \cdot x), \ldots, a(\hat{w}_k \cdot x)] \)
- \( G \) is the function class s.t. \( g(z) = \text{sign}(v \cdot z) \), \( v \in \{-1, 1\}^k \)
- \( \exists f \in F, g_S, g_T \in G \) s.t. \( \max[R_S(g_S \circ f), R_T(g_T \circ f)] \leq \epsilon \)
- Suppose \( \|w_i\| = \|\alpha_i \hat{w}_i - \beta_i w_i\| \) and \( w_i \cdot (\alpha_i \hat{w}_i - \beta_i w_i) = 0 \)
- \( M \) is a full rank \( 2k \times n \) matrix with rows \( w_i, \alpha_i \hat{w}_i - \beta_i w_i \)
- Let \( P_S, P_T \) be distributions on \( X \) with the property \( \forall x, x' \) s.t. \( \|Mx\| = \|Mx'\| \), \( P_T(x) \leq cP_S(x') \) for some \( c \geq 1 \)

**Theorem 2 (\( \omega \) for neural network, fixed representation)**

\[
\omega(R) := cR + \epsilon(1 + c). \ \forall \hat{g}_S \in G, \ \min_{g \in G} R_T(g \circ \hat{f}) \leq \omega(R_S(\hat{g}_S \circ \hat{f})).
\]
Performance guarantees for transferring representations

- Representation fixed by source task
- Neural network example with fixed representation

Compare learning over $G \circ \hat{f}$ to from scratch over $H$ on $T$

Set $\delta = 0.05$, $n = 10$, $k = 5$. Consider the limit

$\epsilon \to 0$, $\hat{R}_S(\hat{g}_S \circ \hat{f}) \to 0$, $m_S \to \infty$, and hence $\omega(\cdot) \to 0$.

We use $d_H = |nk + k| \log |nk + k|$ and $d_G \leq k$
Representation fine-tuned using target task

- \( G \circ \hat{F} := \{ h : \exists f \in \hat{F}, g \in G \text{ s.t. } h = g \circ f \} \), often \( d_{G \circ \hat{F}} = d_H \)
- \( \tilde{h}_{g \circ f} \) is a distribution on \( H \) (i.e. a stochastic hypothesis) corresponding to \( g \circ f \) (e.g. \( g \circ f \) plus noise)
- \( R_T(\tilde{h}) := \mathbb{E}_{x \sim P_T, h \sim \tilde{h}}[h_T(x) \neq h(x)] \), \( \hat{R}_T(\tilde{h}) \) is empirical risk
- Could learn \( T \) from scratch with fixed prior \( \tilde{h}_0 \) and stochastic hypothesis class \( \tilde{H} := \{ \tilde{h}_{g \circ f} : f \in F, g \in G \} \)
- Alternatively, use \( \hat{g}_S \circ \hat{f} \) to construct prior \( \tilde{h}_{\hat{g}_S \circ \hat{f}} \) and stochastic hypothesis class \( \tilde{H}_{G \circ \hat{F}} := \{ \tilde{h}_{g \circ f} : f \in \hat{F}, g \in G \} \)
- PAC-Bayes result bounds generalization error using KL divergence between prior and posterior hypotheses
- Want \( \hat{F} \) ‘small enough’ s.t. \( KL(\tilde{h}||\tilde{h}_{\hat{g}_S \circ \hat{f}}) \leq \omega(R_S(\hat{g}_S \circ \hat{f})) \) \( \forall \tilde{h} \in \tilde{H}_{G \circ \hat{F}} \) for some transferrability function \( \omega \)
- Also want \( \hat{F} \) ‘large enough’ s.t. \( \exists \tilde{h}_{g_T \circ f} \in \tilde{H}_{G \circ \hat{F}} \text{ s.t. } R_T(\tilde{h}_{g_T \circ f}) \leq \epsilon \)
Theorem 3 (Risk upper bound with fine-tuning)

Suppose it is possible to construct $\tilde{H}_{G \circ \hat{F}}$ with the property

$$\forall \tilde{h} \in \tilde{H}_{G \circ \hat{F}}, \quad KL(\tilde{h} || \tilde{h}_{g_s \circ \hat{f}}) \leq \omega(R_S(\hat{g}_S \circ \hat{f})).$$

Then with probability at least $1 - \delta$ over pairs of training sets for $S$ and $T$, $\forall \tilde{h} \in \tilde{H}_{G \circ \hat{F}}$,

$$R_T(\tilde{h}) \leq \hat{R}_T(\tilde{h}) + \sqrt{\omega(\hat{R}_S(\hat{g}_S \circ \hat{f}) + 2\sqrt{2d_H \log \left(2em_S/d_H\right) + 2\log(8/\delta)}} \cdot \frac{2d_H \log \left(2em_S/d_H\right) + 2\log(8/\delta)}{m_S^{m_T} \log 2(m_T - 1) + \log 2m_T/\delta}.$$

- If $\omega(R) = O(R)$, $\hat{R}_S(\hat{g}_S \circ \hat{f})$ is a small constant, and $m_S \gg m_T$, improve on the PAC-Bayes bound for $\tilde{H}$ and $\tilde{h}_0$.

- Neural network with similar assumptions to previous example allows us to define $\omega$ and $\hat{F}$.
Modified regularization penalty

\[
\sum_{i=1}^{m} [-y_i \log \hat{y}_i - (1 - y_i) \log (1 - \hat{y}_i)] + \sum_{j=1}^{l} \left[ \frac{\lambda_1(j)}{2} \| W^{(j)} - \hat{W}^{(j)} \|^2_2 + \frac{\lambda_2(j)}{2} \| W^{(j)} \|^2_2 \right]
\]

- Relax hard constraint on \( \hat{F} \) by using a modified loss function
- Let \( y \) and \( \hat{y} \) be labels and predictions over \( m \) points
- Neural network with \( l \) layers of weights, let \( W^{(j)} \) be the \( j \)th weight matrix and \( \hat{W}^{(j)} \) be its estimate from \( S \)
- Assuming lower level features are more transferrable, \( \lambda_1 \) is a decreasing function
Experiments

- Experiments on MNIST and 20 Newsgroups datasets
- Randomly partition label classes into $S_+$ and $S_-$, $|S_+| = |S_-|
- Construct $T_+$ randomly picking from $S_+$ up to $\gamma := \frac{|S_+ \cap T_+|}{|S_+|}$, then randomly picking from $S_-$ such that $|T_+| = |T_-|
- Let $S$ be the task of distinguishing between $S_+$ and $S_-$ and $T$ be that of distinguishing $T_+$ and $T_-
- \lambda_1(1) = \lambda_2(2) = \lambda := 1$ and $\lambda_1(2) = \lambda_2(1) = 0
- m_T = 500$, $l = 2$, sigmoid activation, average over 10 runs
- MNIST: pixel features, $784 \times 50 \times 1$ network, $m_S = 50000$
- 20 Newsgroups: TF-IDF weighted word frequency features, $2000 \times 50 \times 1$ network, $m_S = 150000$
## Results

<table>
<thead>
<tr>
<th>Technique</th>
<th>MNIST, $\gamma =$</th>
<th>Newsgroups, $\gamma =$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>Base</td>
<td>88.4</td>
<td>87.9</td>
</tr>
<tr>
<td>Fine-tune $\hat{f}$</td>
<td><strong>91.9</strong></td>
<td><strong>93.9</strong></td>
</tr>
<tr>
<td>Fix $\hat{f}$</td>
<td>87.5</td>
<td>92.3</td>
</tr>
<tr>
<td>Fix $\hat{g}_S \circ \hat{f}$</td>
<td>67.4</td>
<td>85.6</td>
</tr>
</tbody>
</table>

- Learn $T$ from scratch (**Base**)
- Transfer $\hat{f}$ from $S$, tune $f$ and train $g$ on $T$ (**Fine-tune $\hat{f}$**)
- Transfer $\hat{f}$ from $S$ and fix, train $g$ on $T$ (**Fix $\hat{f}$**)
- Transfer $\hat{g}_S \circ \hat{f}$ from $S$ and fix (**Fix $\hat{g}_S \circ \hat{f}$**)
Conclusion

- Step towards theoretical foundation for transferring representations, both with and without fine tuning
- Theory motivates transfer regularization penalty to prevent target task overfitting
References


Questions?