A Smoother Way To Train Structured Prediction Models

Overview

- Training structured prediction models is a non-smooth optimization problem involving inference oracles.
- We break the non-smoothness barrier for fast optimization with smooth inference oracles and accelerated incremental algorithms.

Structured prediction

Structured outputs: such as chain of tags in named entity recognition

Score function: \( \phi(x, w) \) measures compatibility of output \( y \) for input \( x \) through

\[
\phi(x, y, w) = \begin{cases} 
\text{high if } y \text{ is a good labeling for } x \\
\text{low if } y \text{ is a poor labeling for } x 
\end{cases}
\]

Inference: finding best output

\[
y^*(x; w) = \arg\max_y \phi(x, y; w)
\]

Given by combinatorial algorithms, e.g., dynamic programming, graph cut/matching

Training: find optimal \( w \) for \( \phi(x, \cdot; w) \), s.t. inference \( y^*(x; w) \) is correct

Gives a loss \( \ell \), use surrogate max-margin loss defined for input-output \( (x, y) \) as

\[
f_i(w) = \max \{ \phi_i(y', w) \}
\]

where

\[
\psi_i(y'; w) = \phi(x, y'; w) + \ell(y, y') - \phi(x, y_i, w)
\]

Optimization problem is

\[
\min_w \left[ F(w) + \frac{1}{n} \sum_i f_i(w) + \frac{\lambda}{2} ||w||_2^2 \right]
\]

where one obtains \( v \in \partial f_i(w) \) by calling inference oracle

\[
\arg\max_y \phi_i(y', w)
\]

Smoothing

Composite objective: Rewrite \( f_i = h(y) g_i \), where

\[
h(z) = \max_{z_0 \in \Delta} \{ z_i \} \quad \text{and} \quad g_i(y') = \{ \phi_i(y'; w) \}_{i \in Y}
\]

Smoothing: Smooth max function \( h(z) = \max_{z_0 \in \Delta} \{ z_i \} - \mu w(u) \) as

\[
h_{\mu}(z) = \max_{z_0 \in \Delta} \{ z_i \} - \mu w(u)
\]

where \( \Delta \subseteq \mathbb{R}^n \) is the simplex, \( \mu > 0 \), and \( w \) is strongly convex

\[
\text{Smoothing type} \quad \mu(u) = \begin{cases} 
\log(\|u\|_2) & \text{log-sum-exp} \\
\log\{\|u\|_2\} & \text{projection on simplex}
\end{cases}
\]

Approximates \( f_i \) to \( O(\mu) \) by smooth max-margin loss

\[
f_{\mu, w} = h_{\mu} \circ g_i
\]

Deep structured prediction

Suppose score given by a learned feature mapping \( \Phi(x, y, w) \), such that

\[
\Phi(x, y, w) = \Phi(x, y_i, w)
\]

is non-linear in \( w \) and the training problem is convex

Idea: Consider linear approximation of \( \psi_i \) around \( z \) as

\[
\psi_i(y, z) = \psi(y, z) + \nabla \psi(y, z)(w - z) \quad \text{and} \quad f_i(w; z) = \max_{w' \in R} \psi_i(y, w, z)
\]

to get a regularized convex model

\[
F_i(w; z) = \frac{1}{n} \sum_i f_i(w; z) + \frac{\lambda}{2} ||w||^2 + \frac{1}{n} ||w - z||^2
\]

Algorithm: At each step \( k \) use convex solver to approximately solve at \( \epsilon \) accuracy

\[
w_{k+1} = \arg\min_w F_{\mu}(w; z_k)
\]

Convergence: Guaranteed to get an \( \epsilon \)-near stationary point after

\[
\mathbb{E}(N) = O\left( \frac{n}{\epsilon^2} \right) \text{ iterations}
\]

Numerical experiments

Pre-defined feature map: convex problem, using structural SVMs

Named entity recognition on CoNLL-2003

Visual object localization on PASCAL VOC

Learned feature map: non-convex problem, using convolutional neural networks

Visual object localization on PASCAL VOC

References