today: structured prediction basics

For example: OCR

\[ x: \text{\underline{B} \underline{C} \underline{A} \underline{E}} \]

\[ y: \text{B R A C E} \]

\[ x = (x_1, \ldots, x_n), \quad x_i \in \{0,1\} \]

\[ \gamma(x) = \gamma(x) \]

\[ \gamma(x) \in \{A, B, \ldots, Z\} \]

\[ \text{let } \gamma \equiv \bigcup_{x} \gamma(x) \]

* task usually gives a structured error \( f \) / loss \( f \)

\[ \ell(y, y^*) = \text{error of predicting } y \text{ when } \text{"truth" is } y \]

[more generally, \( \ell(x, y, y) \)

\[ \ell \text{ "context dependent loss"} \]
More generally, \( O(x, y, y') \) is "context dependent loss".

One example: Hamming loss \( l(y, y') = \sum_{p=1}^{\text{dim.} y} \mathbb{1}_{y_p \neq y'_p} \).

**Learning problem:**

given a training dataset \( D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{\mathcal{N}} \)

goal: learn a prediction mapping \( h_w : X \rightarrow \mathcal{Y} \)

that has low generalization error

\[
\mathbb{L}(w, \mathcal{P}) = \mathbb{E}_{(y, y') \sim \mathcal{P}} \left[ l(y, h_w(x)) \right]
\]

"risk" in ML (Vapnik) / Statistical decision loss

[see lecture 455 of my 86M or review of statistical decision theory]

**Regularized ERM:**

\[
\hat{\mathbb{L}}(w) = \frac{1}{N} \sum_{i=1}^{N} l(y^{(i)}, h_w(x^{(i)})) + R(w)
\]

regularizer

not a.s., non-convergent messy \( \sup \)-hard to minimize

replace it with **surrogate loss** / contrast Fed.

ML / Statistics
\[ \hat{f}(w) = \frac{1}{n} \sum_{i=1}^{n} s(x^{(i)}, y^{(i)}, w) + \theta(w) \]

M-estimator
in statistics

Joint vs Local:

For OCR example:

- Could learn character classifier \( X \rightarrow Y \) directly
- Could learn a correlation model \( P(Y_1, Y_2, \ldots, Y_L) \) [Language model]

Could try to "glue" them together

Vs. structured prediction "end-to-end" training

Other Examples:

1) toy word alignment/matching

\( \mathcal{X} = (x^E_1, \ldots, x^E_L) \) \( \mathcal{Y} = (y^E_1, \ldots, y^E_L) \)

\[ \psi(x) = \sum_{y \in \mathcal{Y}} \mathbb{I}_{y \in \mathcal{Y}} \sum_{i=1}^{L} \mathbb{I}_{y^E_i = y} y_i = 1 \]

\( y_i \leq \sum_{j \in \text{auto}} \hat{y}_{i,j} \leq 1 \)
2) Image segmentation

\[ X = \text{ union of RGB values} \]
\[ L \times L \text{ pixels} \]
\[ S(x) = \{0, 1\}^{L \times L} \]

background vs. object

loss \rightarrow \text{“intersection over union”}

energy-based methods

model \( h_b(z) = \arg \max_{y \in \text{S}(x)} E(x, y, i; w) \)

\[ = \arg \max_{y \in \text{S}(x)} S(x, y; i; w) \]

“score” / “compatibility of \( y \) for \( x \)”

\[ S(x, y; i; w) = -E(x, y; i; w) \]
$S(x, y; w) = -E(x, y; w)$

1. What is $E(x, y; w)$?
   - e.g., $S(x, y; w) = \langle w, \phi(x, y) \rangle$

2. How do you compute $\partial E(x, y; w)$? "Uninhibit" "decoding"
   - Learn from $y(x)$

3. Evaluating $E(x, y; w)$ on training set → surrogate loss $\tilde{E}(w)$

4. How to minimize $\tilde{E}(w)$ to learn $w$?

**Linear case:** $S(x, y; w) = \langle w, \phi(x, y) \rangle$

Let's compare with multiclass setting:

\[ \text{predict argmax } \langle w, \phi(x) \rangle \]

If we define $\phi(x, y) = \begin{pmatrix} 0 \\ x \\ 0 \\ 0 \\ \cdots \end{pmatrix} \in \mathbb{R}^d$

\[ \langle w, \phi(x, y) \rangle = \langle w, \phi(x) \rangle > 0 \] "flat multiclass"

OCR example:

\[ \langle w, \phi(x, y) \rangle = \sum_{i=1}^{L_x} \langle w, \phi(x^i) \rangle \]
OCR example:

\[
\langle w, \Phi(x, y) \rangle = \sum_{p=1}^{Lx} \langle w^{(x)}_{p}, \Phi(x, y_{p}) \rangle + \sum_{p=1}^{Ly} \langle w^{(y)}_{p}, \Phi(x_{p}, y_{p+1}) \rangle
\]

Graphical model:

\[
\begin{array}{cccc}
\quad & y_1 & y_2 & y_3 & y_4 \\
\mathbf{b} & \mathbf{f} & \mathbf{a} & \mathbf{c} & \mathbf{e}
\end{array}
\]

\[\Phi(x_{p}, y_{p}) = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} x_p \cr y_p \cr \vdots \cr y_{p-1} \cr y_{p+1} \end{pmatrix} \in \mathbb{R}^{5x26}
\]

\[
\langle w^{(x)}_{p}, \Phi(x_{p}, y_{p}) \rangle = w_{a} \left[ \phi_{a} \right]
\]

\[
\langle w^{(y)}_{p}, \Phi(x_{p}, y_{p}) \rangle = w_{b} \left[ \phi_{b} \right]
\]

\[\rightarrow \text{ for augmax } S(x, y; w) \text{ use max-sum on chain graphical model}
\]

\[\text{sp. Viterbi algorithm}
\]