(Reranking) A reminder of the learning setting

- $\mathcal{X}$ is a set of possible inputs
- $\mathcal{Y}$ is a set of possible outputs
- A training set $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where each $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$
- We assume that S is generated i.i.d. from an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$
- Goal is to learn a hypothesis function $F : \mathcal{X} \rightarrow \mathcal{Y}$, that minimizes error on the entire distribution $\mathcal{D}$
- E.g., each x_i is a sentence, each y_i is a gold-standard parse

Reranking: Setting

- Three components:
  - ⋆ GEN is a function from a string to a set of candidates
  - ⋆ Φ maps a candidate to a feature vector
  - ⋆ score$(\Phi, x, \hat{y})$ a function that scores the appropriateness of candidate solution $\hat{y}$ for input example x.

- F is of the form: $F(x) = \arg\max_{\hat{y} \in GEN(x)} \text{score}(\Phi, x, \hat{y})$

- For linear classifiers: $F(x) = \arg\max_{\hat{y} \in GEN(x)} \Phi(x, \hat{y}) \cdot \mathbf{w}$, where **w** is a parameter vector.

  I.e., **Choose the highest scoring tree as the most plausible structure**

Reranking: Setting

- In reranking the function GEN gives a set of explicit $\hat{y}$ candidates for each example x, e.g., the list of n-best parses for a sentence x produced by a statistical parser.

- Goal of learning:

  ⋆ Given $\{(x_i, y_i)\}_{i=1}^{n}$, GEN, Φ

  ⋆ How to set **w**?
  to make $\Phi(x, y) \cdot \mathbf{w} \geq \Phi(x, \hat{y}) \cdot \mathbf{w}$, for all candidate solutions $\hat{y}$ produced by GEN

- Reranking is an instance of metalearning

Reranking: training algorithms

- Several algorithms exists for the ranking problem, as variants of the standard learning algorithms: perceptron, boosting, SVMs, etc.

- Instead of correct classification, the learning constraint imposed by ranking (and to be considereded in the objective function) is:

$$\forall \hat{y} \in GEN(x),\ \hat{y} \neq y: \quad \text{score}(\Phi, x, y) > \text{score}(\Phi, x, \hat{y})$$

Reranking: perceptron learning

[Collins and Duffy, 2002]

$S = \{(x_i, y_i)\}_{i=1}^{n}$; assume $GEN(x_i) = \{y_{i1}, \ldots, y_{in_i}\}$

Initialization: set parameters **w** $= 0$
repeat for N epochs
  for $i = 1 \ldots n$
    $j = \arg\max_{j=1\ldots n_i} \Phi(x_i, y_{ij}) \cdot \mathbf{w}$
    if $(y_i \neq y_{ij})$ **then** $\mathbf{w} = \mathbf{w} + \Phi(x_i, y) - \Phi(x_i, y_{ij})$
  end-for
end-repeat
output: **w**

- A simple extension to dual perceptron exists: kernels and voted perceptron can be used

Reranking: Max-margin learning

[Bartlett et al., 2004]

- An iterative algorithm for training a ranking function (based on EG, Exponentiated Gradient optimization)
- Learning bias is to maximize the margin of the training set (SVM-like)

Reranking: Max-margin learning

[Bartlett et al., 2004]

- An iterative algorithm for training a ranking function (based on EG, Exponentiated Gradient optimization)
- Learning bias is to maximize the margin of the training set (SVM-like)
- Specific loss functions can be set
- The algorithm converges to the exact solution
- See slides from Michael Collins' presentation at CoNLL-2006 (PDF file)

Reranking

A couple of technical and practical questions

- How to generate the training set for learning the ranker?
- What if the gold standard y is not among the candidates $\hat{y}$?

Reranking: applications

- Parse reranking

  [Johnson et al, 1999; Collins, 2000; Shen, Sarkar and Joshi, 2003; Riezler et al., 2004; Charniak and Johnson, 2005; Collins and Koo, 2005]

- Reranking for Machine Translation

  [Shen, Sarkar and Och, 2004; Shen and Joshi, 2005]

- Semantic Role Labeling **[Haghighi et al., 2005]**

Reranking: pros & cons

Pros

- Rich complex features can be designed on the complete structure. This may facilitate capturing long-distance dependencies among substructures
- Simple (and efficient) learning algorithms exist

Cons

- Dependence on the base linguistic processor implementing GEN. High recall must be ensured with a few candidates. The theoretical upper bounds on task accuracy can be substantially lowered
- The two-step procedure can make the system less efficient