

Machine learning for sequence alignment

Chuong B. Do
chuongdo@cs.stanford.edu

January 25, 2007

1

Motivation

- computational biology relies on models of *sequence data*
 - proteins (20 different amino acids, physicochemical properties)
 - RNA (single-stranded, complementary base-pairing)
 - DNA (millions of base pairs long)
- this lecture will focus on *protein sequence alignment*
 - lots of applications (e.g., evolutionary analysis, prediction of functional constraints, structure prediction)
 - simple domain, yet the ideas generalize to other sequence models

2

Why machine learning?

- advantages
 - interpretable models
 - parameter estimation
 - inference algorithms
- disadvantages
 - lots of math
 - requires extreme care with implementation

3

Outline

1. traditional methods for sequence alignment
2. modern discriminative learning techniques
 - conditional random fields
 - representation and training
 - algorithms
 - experiments
 - max-margin models
 - training and algorithms
 - experiments
3. what we have not covered

4

Outline

1. **traditional methods for sequence alignment**
2. modern discriminative learning techniques
 - conditional random fields
 - representation and training
 - algorithms
 - experiments
 - max-margin models
 - training and algorithms
 - experiments
3. what we have not covered

5

Definitions

- x and y are two sequences to be aligned
 - x_i is the i th character of x
 - $x_{1:i}$ denotes the substring $x_1x_2 \dots x_i$
 - $|x|$ is the length of x
- \mathcal{A} is the set of all possible alignments of x and y
- score of alignment $a \in \mathcal{A}$ is a summation over columns of
 - substitution scores, $s(x_i, y_j)$
 - gap penalties, $g (< 0)$

6

Needleman-Wunsch review

- find score of optimal alignment via *dynamic programming*

$D(i, j)$ = score of optimal alignment between $x_{1:i}$ and $y_{1:j}$

$$= \max \begin{cases} (i + j) \cdot g & \text{if } i = 0 \text{ or } j = 0 \\ D(i - 1, j - 1) + s(x_i, y_j) & \text{if } i > 0 \text{ and } j > 0 \\ D(i - 1, j) + g & \text{if } i > 0 \\ D(i, j - 1) + g & \text{if } j > 0 \end{cases}$$

- $D(|x|, |y|)$ is the score of the optimal alignment
- use *traceback pointers* to recover alignment (actually, not necessary if you keep the $D(\cdot, \cdot)$ matrix in memory; why?)

7

Alignment as optimization

- score of an alignment $a \in \mathcal{A}$ is a *linear function* of alignment features
- define

$$\mathbf{w} = \begin{bmatrix} s(A, A) \\ s(A, C) \\ \vdots \\ s(Y, Y) \\ g \end{bmatrix} \quad \mathbf{F}(a, x, y) = \begin{bmatrix} \# \text{ of } (A, A) \text{ matches} \\ \# \text{ of } (A, C) \text{ matches} \\ \vdots \\ \# \text{ of } (Y, Y) \text{ matches} \\ \# \text{ of gaps} \end{bmatrix}$$

- find the highest scoring alignment

$$\text{maximize}_{a \in \mathcal{A}} \mathbf{w}^T \mathbf{F}(a, x, y)$$

8

Substitution matrices

- ad hoc methods
 - sequence identity

$$s(x_i, y_j) = 1 \{x_i = y_j\}$$

- weighting transition/transversions (DNA)

	A	G	C	T
A	1	0	-1	-1
G	0	1	-1	-1
C	-1	-1	1	0
T	-1	-1	0	1

9

Substitution matrices

- log-odds methods
 - assume
 - * independence of alignment columns
 - * fixed evolutionary time-scale
 - PAM (Dayhoff 1979)

$$s(x_i, y_j) = \log \left(\frac{\text{probability of } x_i \rightarrow y_j}{\text{background probability of } y_j} \right)$$

- BLOSUM (Henikoff & Henikoff 1992)

$$s(x_i, y_j) = \log \left(\frac{\text{probability of } x_i y_j \text{ in aligned blocks}}{\text{probability of } x_i \times \text{probability of } y_j} \right)$$

10

Gap penalties

- no commonly accepted estimation techniques
- bottom line: guess-and-check!
- what if we want to use affine gap penalties? piecewise linear gap penalties?
- what if we want gap penalties to be context-dependent? (e.g., lower gap penalties in hydrophilic regions of proteins, increase gap penalties in hydrophobic core)

11

Richer feature sets

- what if we knew the secondary structure of the protein (α (alpha helix), β (beta sheet), or γ (coil)) at each position?

$$\mathbf{w} = \begin{bmatrix} s(\mathbf{A}, \mathbf{A}) \\ s(\mathbf{A}, \mathbf{C}) \\ \vdots \\ s(\mathbf{Y}, \mathbf{Y}) \\ g \\ s'(\alpha, \alpha) \\ s'(\alpha, \beta) \\ \vdots \\ s'(\gamma, \gamma) \end{bmatrix} \quad \mathbf{F}(a, x, y) = \begin{bmatrix} \# \text{ of } (\mathbf{A}, \mathbf{A}) \text{ matches} \\ \# \text{ of } (\mathbf{A}, \mathbf{C}) \text{ matches} \\ \vdots \\ \# \text{ of } (\mathbf{Y}, \mathbf{Y}) \text{ matches} \\ \# \text{ of gaps} \\ \# \text{ of } (\alpha, \alpha) \text{ matches} \\ \# \text{ of } (\alpha, \beta) \text{ matches} \\ \vdots \\ \# \text{ of } (\gamma, \gamma) \text{ matches} \end{bmatrix}$$

12

Recap: traditional methods for sequence alignment

- definitions
- review of dynamic programming
- reformulation of alignment as an optimization problem
- parameter estimation
 - substitution matrices
 - gap penalties
 - richer models

13

Outline

1. traditional methods for sequence alignment
2. **modern discriminative learning techniques**
 - conditional random fields
 - representation and training
 - algorithms
 - experiments
 - max-margin models
 - training and algorithms
 - experiments
3. what we have not covered

14

Modern parameter estimation

- given
 - a training set \mathcal{D} of curated alignments (e.g., BAliBASE (Thompson et al. 1994))

$$\mathcal{D} = \left\{ (a^{(i)}, x^{(i)}, y^{(i)}) \right\}_{i=1}^m$$

- goal
 - pose parameter estimation as a convex optimization problem
 - we will focus on two recent formulations
 - * conditional random fields (think logistic regression)
 - * max-margin models (think SVMs)

15

Outline

1. traditional methods for sequence alignment
2. modern discriminative learning techniques
 - **conditional random fields**
 - **representation and training**
 - algorithms
 - experiments
 - **max-margin models**
 - training and algorithms
 - experiments
3. what we have not covered

16

Conditional random fields (CRFs)

- define probabilistic model over alignments

$$\begin{aligned} P(a | x, y; \mathbf{w}) &= \frac{\exp(\mathbf{w}^T \mathbf{F}(a, x, y))}{\sum_{a' \in \mathcal{A}} \exp(\mathbf{w}^T \mathbf{F}(a', x, y))} \\ &= \frac{1}{Z(x, y)} \cdot \exp(\mathbf{w}^T \mathbf{F}(a, x, y)) \end{aligned}$$

where

$$Z(x, y) = \sum_{a' \in \mathcal{A}} \exp(\mathbf{w}^T \mathbf{F}(a', x, y))$$

- also known as conditional *log-linear* models, since

$$\log P(a | x, y; \mathbf{w}) \propto \mathbf{w}^T \mathbf{F}(a, x, y)$$

17

Training CRFs

- maximum (conditional) log-likelihood estimation

$$\underset{\mathbf{w} \in \mathbb{R}^n}{\text{maximize}} \quad \ell(\mathbf{w} : \mathcal{D})$$

where

$$\begin{aligned} \ell(\mathbf{w} : \mathcal{D}) &= \sum_{i=1}^m \log P(a^{(i)} | x^{(i)}, y^{(i)}; \mathbf{w}) \\ &= \sum_{i=1}^m \left[\mathbf{w}^T \mathbf{F}(a^{(i)}, x^{(i)}, y^{(i)}) - \log \left(\sum_{a' \in \mathcal{A}} \exp(\mathbf{w}^T \mathbf{F}(a', x^{(i)}, y^{(i)})) \right) \right] \end{aligned}$$

- note that objective is concave in \mathbf{w}

18

Regularization

- problem
 - recall that $P(a | x, y; \mathbf{w}) \propto \exp(\mathbf{w}^T \mathbf{F}(a, x, y))$
 - overfitting may occur if $|w_k|$ too large; why?
- solution
 - regularized maximum (conditional) log-likelihood estimation

$$\underset{\mathbf{w} \in \mathbb{R}^n}{\text{maximize}} \quad \ell(\mathbf{w} : \mathcal{D}) - C\|\mathbf{w}\|^2$$

- in theory, equivalent to zero-mean Gaussian prior for \mathbf{w}
- in practice, keeps parameters small

19

Numerical optimization

- use *gradient ascent* to optimize $\ell(\mathbf{w} : \mathcal{D}) - C\|\mathbf{w}\|^2$:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \cdot (\nabla_{\mathbf{w}} \ell(\mathbf{w} : \mathcal{D}) - 2C\mathbf{w})$$

where α is the *learning rate*

- iterate until $\|\nabla_{\mathbf{w}} \ell(\mathbf{w} : \mathcal{D}) - 2C\mathbf{w}\| \approx 0$
- better yet, run a second-order method (e.g., conjugate gradients, L-BFGS)

20

Recap: conditional random fields — representation and training

- defined probabilistic model of alignments given sequences
- maximum likelihood training = convex optimization
- regularization to prevent overfitting
- gradient-based optimization

21

Outline

1. traditional methods for sequence alignment
2. modern discriminative learning techniques
 - **conditional random fields**
 - representation and training
 - **algorithms**
 - experiments
 - max-margin models
 - training and algorithms
 - experiments
3. what we have not covered

22

Gradient derivation

- gradient decomposes as sum over training examples

$$\nabla_{\mathbf{w}} \ell(\mathbf{w} : \mathcal{D}) = \begin{bmatrix} \frac{\partial}{\partial w_1} \sum_{i=1}^m \log P(a^{(i)} | x^{(i)}, y^{(i)}; \mathbf{w}) \\ \vdots \\ \frac{\partial}{\partial w_n} \sum_{i=1}^m \log P(a^{(i)} | x^{(i)}, y^{(i)}; \mathbf{w}) \end{bmatrix}$$

- how to compute $\frac{\partial}{\partial w_k} \log P(a | x, y; \mathbf{w})$?
- recall

$$\log P(a | x, y; \mathbf{w}) = \mathbf{w}^T \mathbf{F}(a, x, y) - \log \left(\sum_{a' \in \mathcal{A}} \exp(\mathbf{w}^T \mathbf{F}(a', x, y)) \right)$$

23

Gradient derivation

- to compute gradient,

$$\begin{aligned} & \frac{\partial}{\partial w_k} \left[\mathbf{w}^T \mathbf{F}(a, x, y) - \log \left(\sum_{a' \in \mathcal{A}} \exp(\mathbf{w}^T \mathbf{F}(a', x, y)) \right) \right] \\ &= F_k(a, x, y) - \frac{\sum_{a' \in \mathcal{A}} \exp(\mathbf{w}^T \mathbf{F}(a', x, y)) \cdot F_k(a', x, y)}{\sum_{a'' \in \mathcal{A}} \exp(\mathbf{w}^T \mathbf{F}(a'', x, y))} \\ &= F_k(a, x, y) - \sum_{a' \in \mathcal{A}} \left(\frac{\exp(\mathbf{w}^T \mathbf{F}(a', x, y))}{\sum_{a'' \in \mathcal{A}} \exp(\mathbf{w}^T \mathbf{F}(a'', x, y))} \right) \cdot F_k(a', x, y) \\ &= F_k(a, x, y) - \sum_{a' \in \mathcal{A}} P(a' | x, y; \mathbf{w}) \cdot F_k(a', x, y) \\ &= F_k(a, x, y) - \mathbb{E}_{a'} [F_k(a', x, y)] \end{aligned}$$

24

Stationarity conditions

- at the optimum of the objective function,

$$\nabla_{\mathbf{w}} \ell(\mathbf{w} : \mathcal{D}) = 0$$

- thus, for each k ,

$$\sum_{i=1}^m \left[F_k(a^{(i)}, x^{(i)}, y^{(i)}) - \mathbb{E}_{a'} [F_k(a', x^{(i)}, y^{(i)})] \right] = 0$$

$$\sum_{i=1}^m F_k(a^{(i)}, x^{(i)}, y^{(i)}) = \sum_{i=1}^m \mathbb{E}_{a'} [F_k(a', x^{(i)}, y^{(i)})]$$

25

Gradient computation

- for a single training example (a, x, y) ,

$$\frac{\partial}{\partial w_k} \log P(a, x, y; \mathbf{w}) = F_k(a, x, y) - \mathbb{E}_{a'} [F_k(a', x, y)]$$

- hard part is $\mathbb{E}_{a'} [F_k(a', x, y)]$
- naïve method
 1. enumerate all possible alignments $a' \in \mathcal{A}$ of x and y
 2. compute $\exp(\mathbf{w}^T \mathbf{F}(a', x, y))$
 3. normalize to obtain $P(a', x, y; \mathbf{w})$
 4. compute $\mathbb{E}_{a'} [F_k(a', x, y)]$ explicitly for each k
- what's wrong with this?

26

Efficient gradient computation

- suppose

$$F_k(a', x, y) = \# \text{ of } (A, C) \text{ matches in } a'$$

$$= \sum_{i=1}^{|x|} \sum_{j=1}^{|y|} 1 \{x_i = A \text{ aligns to } y_j = C \text{ in } a'\}$$

- then,

$$\mathbb{E}_{a'}[F_k(a', x, y)] = \sum_{i=1}^{|x|} \sum_{j=1}^{|y|} P_{a'}(x_i = A \text{ aligns to } y_j = C \text{ in } a')$$

$$= \sum_{i=1}^{|x|} \sum_{j=1}^{|y|} \underbrace{P_{a'}(x_i \text{ aligns to } y_j \text{ in } a')}_{\text{use dynamic programming!}} \cdot 1 \{(x_i, y_j) = (A, C)\}$$

27

The “splitting” trick^a

- note that

$$P_{a'}(x_i \text{ aligns to } y_j \text{ in } a')$$

$$= \sum_{a' \in \mathcal{A}} \left(\frac{\exp(\mathbf{w}^T \mathbf{F}(a', x, y))}{Z(x, y)} \right) \cdot 1 \{x_i \text{ aligns to } y_j \text{ in } a'\}$$

- the trick: split a' into two parts:
 - an alignment $a'_f \in \mathcal{A}_f$ of $x_{1:i-1}$ to $y_{1:j-1}$
 - an alignment $a'_b \in \mathcal{A}_b$ of $x_{i:|x|}$ to $y_{j:|y|}$ starting in a (x_i, y_j) match

^aFor more details, see *Biological Sequence Analysis* (Durbin et al. 1999).

The “splitting” trick^a

- then

$$\begin{aligned}
 & \sum_{a' \in \mathcal{A}} \left(\frac{\exp(\mathbf{w}^T \mathbf{F}(a', x, y))}{Z(x, y)} \right) \cdot 1 \{x_i \text{ aligns to } y_j \text{ in } a'\} \\
 &= \frac{1}{Z(x, y)} \cdot \sum_{a'_f \in \mathcal{A}_f} \sum_{a'_b \in \mathcal{A}_b} 1 \{x_i \text{ aligns to } y_j \text{ in } a'\} \\
 & \quad \cdot \exp(\mathbf{w}^T \mathbf{F}(a'_f, x_{1:i-1}, y_{1:j-1})) \cdot \exp(\mathbf{w}^T \mathbf{F}(a'_b, x_{i:|x|}, y_{j:|y|})) \\
 &= \frac{1}{Z(x, y)} \cdot \left(\sum_{a'_f \in \mathcal{A}_f} \exp(\mathbf{w}^T \mathbf{F}(a'_f, x_{1:i-1}, y_{1:j-1})) \right) \\
 & \quad \cdot \left(\sum_{a'_b \in \mathcal{A}_b} \exp(\mathbf{w}^T \mathbf{F}(a'_b, x_{i:|x|}, y_{j:|y|})) \cdot 1 \{x_i \text{ aligns to } y_j \text{ in } a'\} \right)
 \end{aligned}$$

29

The “splitting” trick^a

- define

$$\begin{aligned}
 f(i, j) &= \left(\sum_{a'_f \in \mathcal{A}_f} \exp(\mathbf{w}^T \mathbf{F}(a'_f, x_{1:i-1}, y_{1:j-1})) \right) \\
 b(i, j) &= \left(\sum_{a'_b \in \mathcal{A}_b} \exp(\mathbf{w}^T \mathbf{F}(a'_b, x_{i:|x|}, y_{j:|y|})) \cdot 1 \{x_i \text{ aligns to } y_j \text{ in } a'\} \right)
 \end{aligned}$$

- these are known as the “forward” and “backward” matrices
- then

$$P_{a'}(x_i \text{ aligns to } y_j \text{ in } a') = \frac{1}{Z(x, y)} \cdot f(i, j) \cdot b(i, j)$$

- each can be computed in $O(|x||y|)$ time via dynamic programming; how?

30

Recap: conditional random fields — algorithms

- derived log-likelihood gradient
- analyzed fixed points of log-likelihood
- reduced gradient computation to posterior probability calculation
- showed how to compute posterior probabilities using forward/backward matrices

31

Outline

1. traditional methods for sequence alignment
2. modern discriminative learning techniques
 - **conditional random fields**
 - representation and training
 - algorithms
 - **experiments**
 - max-margin models
 - training and algorithms
 - experiments
3. what we have not covered

32

CONTRAlign (Do et al. 2006)

- Do, C.B., Gross, S.S., and Batzoglou, S. (2006) CONTRAlign: discriminative training for protein sequence alignment. *RECOMB*.
- highlights of results
 - learned parameters (over training sets of size ≈ 40) give performance comparable to the best current aligners using BLOSUM62
 - easy to incorporate extra external information (e.g., solvent accessibility, secondary structure)
 - hydrophathy-dependent gap penalties give significant boost in performance

33

Outline

1. traditional methods for sequence alignment
2. modern discriminative learning techniques
 - conditional random fields
 - representation and training
 - algorithms
 - experiments
 - **max-margin models**
 - **training and algorithms**
 - experiments
3. what we have not covered

34

Introduction to max-margin models

- differences from CRFs
 - no direct probabilistic interpretation
 - formulate optimization problem directly
 - solutions based on quadratic or linear programming
- if you know support vector machines (SVMs)
 - SVM : classification :: max-margin models : sequence labeling

35

Optimization problem

- define

$$\Delta(a^{(i)}, a') = \text{loss for choosing } a' \text{ when true alignment is } a^{(i)}$$

- max-margin estimation

$$\begin{aligned} & \underset{\mathbf{w} \in \mathbb{R}^n, \boldsymbol{\xi} \in (\mathbb{R}^+)^m}{\text{minimize}} && C \|\mathbf{w}\|^2 + \sum_{i=1}^m \xi_i \\ & \text{subject to} && \mathbf{w}^T \mathbf{F}(a^{(i)}, x^{(i)}, y^{(i)}) \geq \\ & && \mathbf{w}^T \mathbf{F}(a', x^{(i)}, y^{(i)}) + \Delta(a^{(i)}, a') - \xi_i \\ & && a' \in \mathcal{A}, i = 1, \dots, m \end{aligned}$$

- **Proposition.** For any optimal solution $(\mathbf{w}^*, \boldsymbol{\xi}^*)$ of the above optimization problem, ξ_i^* is an upper-bound on the loss for training example $(a^{(i)}, x^{(i)}, y^{(i)})$ when using parameters \mathbf{w}^* ; why?

36

Training

- max-margin estimation

$$\begin{aligned} & \underset{\mathbf{w} \in \mathbb{R}^n, \boldsymbol{\xi} \in (\mathbb{R}^+)^m}{\text{minimize}} && C \|\mathbf{w}\|^2 + \sum_{i=1}^m \xi_i \\ & \text{subject to} && \mathbf{w}^T \mathbf{F}(a^{(i)}, x^{(i)}, y^{(i)}) \geq \\ & && \mathbf{w}^T \mathbf{F}(a', x^{(i)}, y^{(i)}) + \Delta(a^{(i)}, a') - \xi_i \\ & && a' \in \mathcal{A}, i = 1, \dots, m \end{aligned}$$

- why not just solve this directly?
- what if we just want an approximate solution?

37

Generic cutting plane algorithm

- S = set of active constraints (initially empty)
 1. solve optimization problem under constraint set S
 2. look at *original* problem constraints and identify most violated constraint c
 - (a) if no violated constraints, terminate
 - (b) otherwise, $S \leftarrow S \cup \{c\}$, and go to step 1

38

Generic cutting plane algorithm

- key intuitions
 - gradually reduce space of feasible solutions (new constraints = cutting planes)
 - require each constraint to be satisfied within a tolerance of ϵ :
$$\mathbf{w}^T \mathbf{F}(a^{(i)}, x^{(i)}, y^{(i)}) \geq \mathbf{w}^T \mathbf{F}(a', x^{(i)}, y^{(i)}) + \Delta(a^{(i)}, a') - \xi_i - \epsilon$$
 - only a polynomial number of constraints needed! (Tsochantaridis et al. 2004)

39

Cutting plane for sequence alignment

- how to identify most violated constraint?
 - let $(\mathbf{w}, \xi) =$ solution under constraint set S
 - a constraint associated with $a' \in \mathcal{A}$ is approximately satisfied if
$$\mathbf{w}^T \mathbf{F}(a^{(i)}, x^{(i)}, y^{(i)}) \geq \mathbf{w}^T \mathbf{F}(a', x^{(i)}, y^{(i)}) + \Delta(a^{(i)}, a') - \xi_i - \epsilon$$
 - so a violation occurs when
$$0 \leq \mathbf{w}^T \mathbf{F}(a', x^{(i)}, y^{(i)}) - \mathbf{w}^T \mathbf{F}(a^{(i)}, x^{(i)}, y^{(i)}) + \Delta(a^{(i)}, a') - \xi_i - \epsilon$$
 - the most violated constraint corresponds to

$$\arg \max_{a' \in \mathcal{A}} \left(\mathbf{w}^T \mathbf{F}(a', x^{(i)}, y^{(i)}) + \Delta(a^{(i)}, a') \right)$$

40

Decomposable loss functions

- to find most violated constraint, solve

$$\arg \max_{a' \in \mathcal{A}} \left(\begin{bmatrix} \mathbf{w} \\ 1 \end{bmatrix}^T \begin{bmatrix} \mathbf{F}(a', x, y) \\ \Delta(a, a') \end{bmatrix} \right)$$

- in certain cases, this is easy!
 - example

$$\begin{aligned} \Delta(a, a') &= \text{number of matched residues in } a \text{ missing from } a' \\ &= \text{number of matched residues in } a \\ &\quad - \text{number of matched residues common to } a \text{ and } a' \end{aligned}$$

41

Loss-augmented inference

- modify Needleman-Wunsch recurrences:

$$D'(i, j) = \max \begin{cases} (i + j) \cdot g & \text{if } i = 0 \text{ or } j = 0 \\ D'(i - 1, j - 1) + s(x_i, y_j) & \text{if } i > 0 \text{ and } j > 0 \\ \quad - 1 \{x_i \text{ aligns with } y_j \text{ in } a\} \\ D'(i - 1, j) + g & \text{if } i > 0 \\ D'(i, j - 1) + g & \text{if } j > 0 \end{cases}$$

- then,

$$\begin{aligned} &D'(|x|, |y|) + \text{number of matched residues in } a \\ &= \max_{a' \in \mathcal{A}} \left(\mathbf{w}^T \mathbf{F}(a', x^{(i)}, y^{(i)}) + \Delta(a^{(i)}, a') \right) \end{aligned}$$

42

Solving optimization problem for a specific constraint set

- how to solve

$$\begin{aligned} & \underset{\mathbf{w} \in \mathbb{R}^n, \xi \in (\mathbb{R}^+)^m}{\text{minimize}} && C \|\mathbf{w}\|^2 + \sum_{i=1}^m \xi_i \\ & \text{subject to} && \mathbf{w}^T \mathbf{F}(a^{(i)}, x^{(i)}, y^{(i)}) \geq \\ & && \mathbf{w}^T \mathbf{F}(a', x^{(i)}, y^{(i)}) + \Delta(a^{(i)}, a') - \xi_i \\ & && \text{for all constraints belonging to } S \end{aligned}$$

when $|S|$ is not too big?

- standard SVM problem!

43

Recap: max-margin methods — training and algorithms

- set up optimization with exponentially many constraints
- cutting-plane algorithm
- find most violated constraint
- solve reduced optimization problem

44

Outline

1. traditional methods for sequence alignment
2. modern discriminative learning techniques
 - conditional random fields
 - representation and training
 - algorithms
 - experiments
 - max-margin models
 - training and algorithms
 - **experiments**
3. what we have not covered

45

Structural SVMs (Yu et al. 2006)

- Yu, C.-N., and Joachims, T. (2006) Training protein threading models using structural SVMs. *ICML Workshop on Learning in Structured Output Spaces*.
- also to appear in RECOMB 2007
- highlights of results
 - addresses problem of threading sequence onto a protein of known structure
 - outperforms SSALN (Qiu and Elber 2006), the current best approach (which is in turn more accurate than GenThreader (Jones 1999) or FUGUE (Shi et al. 2001))

46

Recap: modern discriminative learning techniques

- two techniques for parameter learning in sequence-based models
 - conditional random fields
 - max-margin models
- derived the main algorithms
 - CRFs: training → numerical optimization
 - max-margin models: training $\xrightarrow{\text{cutting plane}}$ sequence of SVM problems
- two instances of these models in practice

47

Outline

1. traditional methods for sequence alignment
2. modern discriminative learning techniques
 - conditional random fields
 - representation and training
 - algorithms
 - experiments
 - max-margin models
 - training and algorithms
 - experiments
3. **what we have not covered**

48

What we have not covered

1. feature construction for alignment algorithms
2. implementation efficiency details
3. validation and training set construction
4. multiple alignment
5. protein structural alignment
6. RNA alignment
7. DNA alignment

Final point: each of the learning algorithms described today is generic — pick your favorite problem and apply it!