Topic for today: gene prediction


“We describe CRAIG, a new program for ab initio gene prediction based on a conditional random field model with semi Markov structure that is trained with an online large-margin algorithm related to multiclass SVMs.”
General perspective on problem

- General problem of non-hierarchical labeled bracketing
  - e.g., sentence segmentation; named entity extraction
- Given an input sequence from a vocabulary $\Sigma_1$
  - e.g., Bill Gates works at Microsoft
- Find an output sequence from a vocabulary $\Sigma_2$
  - e.g., Bill:B$_p$ Gates:I$_p$ works:O at:O Microsoft:B$_c$
- Label set corresponds to bracketing
  - (Bill Gates)$_{person}$ works at (Microsoft)$_{company}$
Application to Gene Prediction

• Of key importance are features of relevance to problem
  – For English Named Entities, capitalization
  – For Gene prediction, start and stop codons

• Given labeled training data, need a learning algorithm, to define the best scoring output sequence for a given input sequence

• Need an efficient inference algorithm, to find the best scoring output sequence for a given input sequence

• Hidden Markov models (HMMs) are one solution
  – related modeling approaches are yielding improvements
Agenda

• Introduction to gene prediction task

• Brief presentation of HMMs for this task
  – More complicated HMMs mirroring finer gene structure

• Discriminative versus Generative models

• Discriminatively trained Markov models
  – Conditional random fields (CRF)

• More advanced model structures and learning approaches
  – e.g., Semi-Markov models and on-line large margin algorithms
Gene structure
Input and output sequences

- Gene structure in eukaryotes (complex cell structures)
  - exons (translated regions) and introns (intervening sequences)
- Input nucleotide sequences: $\Sigma_1 = \{A,C,T,G\}$
- Output begin, inside and outside labels for exons, introns and intergenic regions: $\Sigma_2 = \{B_e, I_e, B_i, I_i, O\}$ or $\Sigma_2 = \{E, I, O\}$
- Minor complication number 1: codons
  - In exons, three nucleotides correspond to a target amino acid
  - Depending on reading frame, could start at $j$, $j+1$ or $j+2$
  - Start codon: ATG; stop codons: TAG, TGA, TAA
**Ab initio gene prediction**

- Use only *intrinsic* evidence for gene prediction
  - e.g., presence of start and stop codons
- In contrast to *extrinsic* evidence from similar sequences
  - Overlap with known mRNA sequences
  - Tends to be high specificity, low sensitivity
- In fact, this is a fuzzy distinction
  - Stochastic models like HMMs use features to define likely (high probability) structures
  - Typically trained from labeled data
  - Hence “extrinsic” evidence could be thought of as a feature
What is the task?

• ... and how do we evaluate?

• Consider a simple (fake) example with a single exon

• Given an input, label the gene regions on the sequence:
  example input: GCATGGTCAGTTAGGAG
  example output: OOEEEEEEEEEEEEEEOOO
  labeled bracketing: GC (E ATGGTCAGTTAG) GAG

• Typically evaluated at three different levels:
  – Base accuracy: coding versus non-coding classification
  – Exon accuracy: found exons versus reference exons
  – Gene accuracy: complete gene labeling versus reference
Evaluation metrics

• Base accuracy: coding versus non-coding nucleotide

Sensitivity (Sn) = \frac{TP}{TP + FN} \quad \text{Specificity (Sp)} = \frac{TN}{TN + FP}

• Exon accuracy: start and end positions of exons

Sn = \frac{\# \text{ of correct exons}}{\# \text{ of actual exons}} = \frac{c}{g} = \text{Recall}

Sp = \frac{\# \text{ of correct exons}}{\# \text{ of guessed exons}} = \frac{c}{t} = \text{Precision}

F-measure = \frac{2Sn \cdot Sp}{Sn + Sp} = \frac{2c}{g + t}

• Gene-level accuracy: totally correct genes / number of genes
Hidden Markov models

- Sequence of hidden states, representing variables $X$
  - e.g., whether or not in an Exon region
- States output the observed values $y$
  - in this case, the particular nucleotide
- Typical graphical model representation:
HMM parameterization

• Model consists of two kinds of parameters
  
  – Transition probabilities between states:
    \[ P(X_i = x \mid X_{i-1} = x') \], for some instantiated values \( x, x' \)
  
  – Emission probabilities from states to observations:
    \[ P(y_i \mid X_i = x) \]

• When we include start and end states, this defines a probability distribution over joint state/observation sequences
  
  – Can use it to infer the “best” state sequence for a given observation sequence
Main issue: define states

- For current problem (single exon gene) start with ‘E’ and ‘O’
- These are hidden states – don’t know the state of each symbol
- Recall first complication: codons
  - Can define multiple ‘E’ states corresponding to start/stop:
    - $E_0 =$ first nucleotide of start (followed by $E_1, E_2$)
    - $E_4 =$ first nucleotide of stop (followed by $E_5, E_6$)
  - Single start codon, just 3 possible stop codons
- To make topology explicit, need to define finite-state automaton
Explicit finite state automaton topology

- $\epsilon$ transitions are from state-to-state (no output)
- Observation probabilities from ‘O’ state $\lambda_y$ for $y \in \{A,C,T,G\}$
  - Conditional probabilities out of states, i.e., $\sum \lambda_y = 1$
  - Multiply probabilities along paths
- State $E_0$ must output symbol A and transition to state $E_1$
Explicit finite state automaton topology (remove $\epsilon$)
Extending HMMs

• Can capture richer dependencies by increasing number of states
• Certain regions of exons or introns may have higher expectations for certain kinds of sub-sequences
• To maintain HMM model, still define in terms of
  – Transition probabilities from state-to-state
  – Emission or observation probabilities of symbols given states
• Efficient parameter estimation techniques
  – Expectation Maximization (EM)
• Efficient inference algorithms (Viterbi)
Exons, introns and splice sites
HMM sub-model areas

- Break sequence into regions (states)
- Create HMM out of state transition probabilities and symbol emission probabilities
- From Augustus (Stanke and Waack, 2003):
Multiple exon genes

- Genes with multiple exons have introns
- Introns are non-coding, hence reading frame can shift from exon to exon
- Exons are relatively constrained in length
- Introns can be very long
  - “Duration” modeling is a key consideration
    * Similar to HMMs when used to model speech, which can vary in duration
  - Brute force length modeling computationally expensive
Duration modeling

• Problem: to give a score specifically to sequence of length $k$, need a state that means ‘length $k$’
  – Large flexibility in modeling comes at computational cost
• Similar problems in approximate string matching with gaps
  – Linear gap models versus affine gap models
• Augustus solution: specific length up to a point, then parametric
  – Introns begin with specific probabilities of specific lengths
  – Then one symbol at a time at a particular cost (looping)
Multiple exon genes (Stanke and Waack, 2003)
HMM structure (forward strand, Stanke and Waack)
HMM structure (reverse strand, Stanke and Waack)
Discriminative versus Generative models

- HMMs are examples of joint, generative models
  - Provide joint probability of both input and output, $p(x, y)$
  - Typically trained to maximize joint likelihood of training set

- Joint likelihood, however, is not particularly well correlated with system objectives, like accuracy of gene prediction
  - Want conditional likelihood of output given input, $p(y \mid x)$
  - Estimate model parameters so that for “bad” outputs conditional likelihood lower than for “good” outputs

- Discriminative models are trained to discriminate good from bad
Example input: A B C

Ranked list of possible outputs:

- A:X B:Y C:Z D:W
- A:V B:Y C:Z D:W
- A:X B:V C:Y D:W
- A:X B:Y C:Y D:W
- A:V B:Y C:Y D:W
- A:V B:Y C:Y D:W

- We have 6 output candidates for the given input
- Suppose the red candidate is true, want to move it up the list
- Can reward or penalize “features” of candidates
- Some features (e.g., presence of D:W) may help joint likelihood but won’t change ranking
Log linear models

- Assume that we are given
  - A training corpus: \((x^{(t)}, y^{(t)})\) for \(1 \leq t \leq T\) examples
    (input sequences and their reference output labels)
  - A mechanism for identifying all candidate outputs for a given input \(x\),
    typically denoted \(\text{GEN}(x)\)
  - A mapping from input/output sequences to a feature vector \(\phi(x, y)\)
  - A parameter vector \(\alpha\) of the same dimension \((d)\) as \(\phi\)

- The score of an input/output sequence \((x, y)\) is the dot product of \(\phi\) and \(\alpha\)
  \[
  S_\alpha(x, y) = \phi(x, y) \cdot \alpha = \sum_{i=1}^{d} \phi_i(x, y) \alpha_i
  \]

- The “best” output is the one that scores the highest
  \[
  \hat{y}(x, \alpha) = \operatorname{argmax}_{y \in \text{GEN}(x)} S_\alpha(x, y)
  \]
Main questions of approach

• What is the GEN function?
  – Could be a simpler model, which gives its $k$-best
  – For gene prediction, can use dynamic programming to explore all possible outputs, given the tag set

• What are the features of the model?
  – To be discriminative, need at least one output label
  – To use dynamic programming, must constrain number of output labels in a single feature
  – Typically sequences of input and output symbols

• How do we set the parameter values in $\alpha$?
Perceptron and Conditional Random Fields

- Very popular and effective perceptron algorithm works as follows:
  - Initialize $\alpha$ to all zeros
  - At each example $(x^{(t)}, y^{(t)})$ find $\hat{y}(x, \alpha)$
  - Then update $\alpha$ to $\alpha'$ as follows
    \[
    \alpha' = \alpha + \phi(x^{(t)}, y^{(t)}) - \phi(x^{(t)}, \hat{y}(x, \alpha))
    \]
  - Move on to next example, using $\alpha'$; iterate through $T$ examples
- One example at a time, then update, means the algorithm is online
- Conditional Random Fields (CRF) has different objective function
  - Maximize conditional probability of reference output
• Calculate the scores of each candidate and normalize over list
  – Make exponentials of scores sum to 1
• This defines a global conditional distribution $p(y \mid x)$
• Try to give as much of this conditional probability mass to the reference as possible
“We describe CRAIG, a new program for ab initio gene prediction based on a conditional random field model with semi-Markov structure that is trained with an online large-margin algorithm related to multiclass SVMs.”

Still to do: large-margin, multiclass SVMs and semi-Markov
But first: why bother (what’s the payoff)?
Major benefits of CRAIG approach

- Ability to easily combine many disparate features (unlike HMM)
  - Basic sub-sequence type features as described earlier
  - Features characterizing expected sub-sequences in regions
  - Features characterizing transitions between regions
  - Scores from other models, e.g., windowed weight array models (WWAMS)
  - Could include scores from other competitor models
  - Could even include arbitrary homology features

- Global rather than local optimization of parameters
- Similar graph structure to HMMs
- Achieves significant improvements over state-of-the-art competitors such as Augustus, GenScan (vanilla and ++) and Genezilla
  - particularly with lengthy introns
FST structure of CRAIG model
### Some results on ENCODE294

<table>
<thead>
<tr>
<th>Level</th>
<th>Augustus F-measure</th>
<th>CRAIG F-measure</th>
<th>Absolute Improvement</th>
<th>Relative Error reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>76.6</td>
<td>82.6</td>
<td>6.0</td>
<td>25.6</td>
</tr>
<tr>
<td>Exon All</td>
<td>57.3</td>
<td>66.2</td>
<td>8.9</td>
<td>20.8</td>
</tr>
<tr>
<td>Initial</td>
<td>36.3</td>
<td>44.5</td>
<td>8.2</td>
<td>12.9</td>
</tr>
<tr>
<td>Internal</td>
<td>66.0</td>
<td>76.2</td>
<td>10.2</td>
<td>30.0</td>
</tr>
<tr>
<td>Terminal</td>
<td>41.2</td>
<td>40.8</td>
<td>-0.4</td>
<td>-0.7</td>
</tr>
<tr>
<td>Single</td>
<td>32.3</td>
<td>35.9</td>
<td>3.6</td>
<td>5.3</td>
</tr>
<tr>
<td>Gene-level</td>
<td>19.2</td>
<td>25.1</td>
<td>5.9</td>
<td>7.3</td>
</tr>
</tbody>
</table>
Discussion

- Overall, these are large accuracy improvements

- Paper doesn’t tease apart what the improvements are due to
  - Discriminative estimation
  - Novel features or feature combinations
  - Training data preparation (simulation of long intergenic regions)

- What is most promising is the flexibility for incorporating new, novel features
  - Don’t have to worry about modeling constraints as with HMMs
  - Still, for dynamic programming, need local features
Semi-Markov modeling

- Most shallow bracketing/segmentation perform inference via tags
  - e.g., \{E,I,O\}

- These correspond to brackets, e.g.,

- Might want features that are
  up at the bracket level

- Standard approach for
  this has $N^2$ complexity
  for strings of length $N$

- Semi-Markov models limit the length of constituents to $L$
  - Complexity is $LN$ instead of $N^2$ (ENCODE294 avg. $> 28k$)
Within classification, *large margin* learning techniques try to find a parameterization of the model that maximizes inter-class margin.
Large margin models

• Trying to classify the “good one” versus the rest

• In the current case, the margin is loss sensitive
  – “Worse” candidates should have more difference between their score and the score of the reference than “better” candidates
  – Need to define loss $L(y^{(t)}, \hat{y})$ for best scoring candidate $\hat{y}$
  – In the current paper, they use a loss based on the correlation coefficients at the base level: high correlation, low loss

• Support Vector Machines (SVMs) are a common large margin learning approach

• Multiclass SVMs can handle multiple classes, rather than just two
CRAIG model specifics

• Inference: semi-Markov
• Features: linear model, segment features, sequence patterns, distributional features (from generative models)
  – Binning of features
• Learning: online, similar to perceptron (MIRA)
  – update based on reference and 1-best
  – find parameters at each step that enforce a loss-sensitive margin (minimal difference from previous step)
  – Use averaging to avoid overfitting
• Some simulated training data (long intergenic regions)
Topic for today: gene prediction


“We describe CRAIG, a new program for ab initio gene prediction based on a conditional random field model with semi-Markov structure that is trained with an online large-margin algorithm related to multiclass SVMs.”