Forecasting and Simulation **Summer Term 2024**

`http://ml3.leuphana.de/lectures/summer24/FS`
Machine Learning Group, Leuphana University of Lüneburg

Soham Majumder (soham.majumder@leuphana.de)

Exercise 7: Conditional Random Fields

1. CRF - Partition Function

Markov Random Field (MRF) models the joint probability $P(\mathbf{x}, \mathbf{y})$ of an input $\mathbf{x}$ and a corresponding output $\mathbf{y}$, where both $\mathbf{x}$ and $\mathbf{y}$ are sets of random variables with some interdependencies among them. This probability is usually given as

$$P(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle),$$

where $\Phi(\mathbf{x}, \mathbf{y})$ is a vector of global features that model dependencies among the variables, $\mathbf{w}$ is the vector of model parameters, $\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$ is the dot product between $\mathbf{w}$ and $\Phi(\mathbf{x}, \mathbf{y})$, and Z is the partition function. In order to guarantee that $P(\mathbf{x}, \mathbf{y})$ is a probability distribution, the partition function must be:

$$Z = \sum_{\mathbf{x}'} \sum_{\mathbf{y}'} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}', \mathbf{y}') \rangle),$$

i.e. the sum of the log-linear values for all possible pairs $\mathbf{x}', \mathbf{y}'$ of input-output variables.

Conditional Random Field (CRF), in the other hand, takes a discriminative approach and models the conditional probability $P(\mathbf{y}|\mathbf{x})$ directly, i.e. it is assumed that the input $\mathbf{x}$ is always given. Considering this model, we have that:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle),$$

where $Z(\mathbf{x})$ is the conditional partition function for the given input $\mathbf{x}$.

Using the equations above and the two basic rules of probability, show that:

$$Z(\mathbf{x}) = \sum_{\mathbf{y}'} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}') \rangle).$$

Hint:	Sum rule:	$P(x) = \sum_{y} P(x, y)$
	Product rule:	$P(x, y) = P(y\|x)P(x)$

Solution

By the product rule, we have that:

$$P(\mathbf{y}|\mathbf{x}) = \frac{P(\mathbf{x}, \mathbf{y})}{P(\mathbf{x})} = \frac{1}{Z} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle) \frac{1}{P(\mathbf{x})}.$$

Then, by the sum rule, we have that:

$$\begin{aligned} P(\mathbf{x}) &= \sum_{\mathbf{y}'} P(\mathbf{x}, \mathbf{y}') \\ &= \sum_{\mathbf{y}'} \frac{1}{Z} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}') \rangle) \\ &= \frac{\sum_{\mathbf{y}'} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}') \rangle)}{Z}. \end{aligned}$$

Finally, substituting $P(\mathbf{x})$ in the first equation, we have that:

$$\begin{aligned} P(\mathbf{y}|\mathbf{x}) &= \frac{1}{Z} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle) \frac{1}{P(\mathbf{x})} \\ &= \frac{1}{Z} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle) \frac{Z}{\sum_{\mathbf{y}'} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}') \rangle)} \\ &= \frac{\exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle)}{\sum_{\mathbf{y}'} \exp(\langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}') \rangle)}. \end{aligned}$$

2. CRF – Feature Templates

One of the main advantages of linear-chain CRF over HMM is that the former can easily handle arbitrary features $\Phi(\mathbf{x}, \mathbf{y}) = (\phi_i(\mathbf{x}, \mathbf{y}))$. These features represent the interdependencies among input $\mathbf{x}$ and output $\mathbf{y}$. For sequence labeling, where $\mathbf{x} = (x_1, \ldots, x_T)$ and $\mathbf{y} = (y_1, \ldots, y_T)$, it is very common, inspired by HMM, to use two sets of features: transition features $\phi_i^{\text{trans}}(y_t, y_{t-1})$ and observation features $\phi_i^{\text{obs}}(y_t, \mathbf{x})$. Moreover, these features are usually *binary*, i.e., for a specific sequence $(\mathbf{x}, \mathbf{y})$ and time-step t, an observation feature is active (function $\phi_i^{\text{obs}}(y_t, \mathbf{x}) = 1$) or not (function $\phi_i^{\text{obs}}(y_t, \mathbf{x}) = 0$). The transition features are also binary, i.e., $\phi_i^{\text{trans}}(y_t, y_{t-1}) \in \{0, 1\}$.

The system designer has a lot of freedom on how to define feature functions, but a common approach is to make use of the so called *feature templates*[*] in order to instantiate features given a dataset of sequences $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\}$. For instance, we can define the following feature templates:

$$\begin{aligned} \langle y_t, y_{t-1} \rangle & \quad \text{(transition features)} \\ \langle y_t, x_t \rangle & \quad \text{(observation features)} \end{aligned}$$

Instantiate these feature templates for the input sequence $\mathbf{x} = (\text{We, are, clever})$ and considering two possible states: $\mathcal{Y} = \{1, 2\}$. Instantiate features for all possible outputs, i.e., $y_t = 1$ and $y_t = 2$, for $t = 1, \ldots, T$.

[*] You can learn more about feature templates in Sections 5.2.2 (`https://web.stanford.edu/~jurafsky/slp3/5.pdf`) and 8.5.1 (`https://web.stanford.edu/~jurafsky/slp3/8.pdf`) from Jurafsky's NLP book (draft of 3rd edition: `https://web.stanford.edu/~jurafsky/slp3/`).

Solution

Refer to Ex 7 solutions.pdf

3. Structured Perceptron for CRF

Remember: Given an input sequence $\mathbf{x} = (x_1, x_2, \dots, x_T)$ and HMM parameters π, a, b, the log-based Viterbi algorithm finds $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_T) = \arg\max_{y'} P(\mathbf{y}'|\mathbf{x})$. Pseudo-code from our solution to Exercise 5 (for details regarding the notation please look at the notebook) is given below.

```
• for $i = 1, \dots, N$
    ◦ $\log \delta_1(i) = \log \pi_i + \log b_i(x_1)$
    ◦ $\psi_1(i) = -1$
• for $t = 2, \dots, T$
    ◦ for $j = 1, \dots, N$
        ∗ $\log \delta_t(j) = \max_i(\log \delta_{t-1}(i) + \log a_{ij}) + \log b_j(x_t)$
        ∗ $\psi_t(j) = \arg\max_i(\log \delta_{t-1}(i) + \log a_{ij})$
• $\hat{y}_T = \arg\max_i \delta_T(i)$
• for $t = T-1, \dots, 1$
    ◦ $\hat{y}_t = \psi_{t+1}(\hat{y}_{t+1})$
```

Consider a CRF model with parameters $\mathbf{w} = (\mathbf{w}^{\text{trans}}, \mathbf{w}^{\text{obs}})$, where $\mathbf{w}^{\text{trans}} = (w_1^{\text{trans}}, \dots, w_{d_{\text{trans}}}^{\text{trans}})$ and $\mathbf{w}^{\text{obs}} = (w_1^{\text{obs}}, \dots, w_{d_{\text{obs}}}^{\text{obs}})$ are, respectively, the transition and observation weights. Consider also the transition feature functions $\phi_j^{\text{trans}}(y_t, y_{t-1})$, for $j = 1, \dots, d_{\text{trans}}$; and the observation feature functions $\phi_j^{\text{obs}}(\mathbf{x}, y_t)$, for $j = 1, \dots, d_{\text{obs}}$. Then, for a given pair of input-output sequences $(\mathbf{x}, \mathbf{y})$, the conditional log-probability $P(\mathbf{y}|\mathbf{x})$ under this model is given by:

$$\log P(\mathbf{y}|\mathbf{x}) = \sum_{t=2}^{T} \sum_{k=1}^{d_{\text{trans}}} w_k^{\text{trans}} \cdot \phi_k^{\text{trans}}(y_t, y_{t-1}) + \sum_{t=1}^{T} \sum_{k=1}^{d_{\text{obs}}} w_k^{\text{obs}} \cdot \phi_k^{\text{obs}}(\mathbf{x}, y_t) - \log Z(\mathbf{x}).$$

Now, perform the following task:

Adapt the algorithm above to find the most probable output sequence $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}'} \log P(\mathbf{y}'|\mathbf{x})$ for a given CRF model and input sequence $\mathbf{x}$. Basically, you need to replace the terms $\log a_{ij}$ and $\log b_j(x_t)$ by sums overs the transitions and observations features, respectively. You should observe that the considered linear-chain feature functions factor into the equation the same way as the HMM parameters a and b (i.e. two consecutive output tags and the current output tag, respectively). You can also ignore the start probabilities π from the HMM, since they are not defined in the above CRF model (although we could do it).

Hint: You can use Section 8.5 of Jurafsky's book as a reference to solve this task. However, try to not just translate Equation 8.33 to our notation. Try to understand the adaptations and be able to reason about them in class. If you struggle with understanding, we suggest reading up on the max-sum (and sum-product) inference algorithms for graphical models as found for example in [Bishop, 2006], Chapter 8.4.

You may also want to work on the following bonus tasks:

(i) Bonus task 1: Consider the input sequence $\mathbf{x}$ = (We, are, clever) and the defined features from Task 2. Set $\mathbf{w} = (\mathbf{w}^{\text{trans}}, \mathbf{w}^{\text{obs}})$ to arbitrary values and execute (manually) the Viterbi algorithm to find $\hat{y} = \arg\max_{\mathbf{y}'} P(\mathbf{y}'|\mathbf{x})$ under the resulting CRF model.

(ii) Bonus task 2: Revisit [Altun et al., 2003] and try to connect what you have just learned to the efficiency argument in Section 4.

Solution

Refer to Ex 7 solutions.pdf for the main task.

4. Conditional Random Fields for NER

This is a continuation of the NER task of Exercise 5.

Use the provided notebook *exercise-06-CRF.ipynb* to train a CRF model for NER. You will need to finish the implementation (look at the `#TODO` comments). Experiment with different observation features (see the `encode_sentence(...)` method) in order to get the best performance on the DEV set. Remember that you can use any input features (word, POS tag, chunk tag) from any token to define observation features for one token x_t. Observe that we use just one feature encoder (`x_enc`) for all observation features. It means that in case you use the same input feature (let us say *word*) from different tokens (x_t and the next token x_{t+1}) you should prefix the feature name with different strings (e.g. `word=xxx` and `word+1=xxx`), so that the model treats the features differently.

Solution

See *solution-07-CRF.ipynb*.

References

[Altun et al., 2003] Altun, Y., Tsochantaridis, I., and Hofmann, T. (2003). Hidden markov support vector machines. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 3–10.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning.* Springer.