An Exploration of Transaction Set Identification

Hye Woong Jeon

Abstract

This paper introduces a Markov chain method for identifying electronic data interchange (EDI) transaction sets. With EDI being an older system, EDI files may not strictly adhere to the official transaction set standards. Correctly identifying transaction sets will effectively parse the intent of corrupted EDI files, hence making transaction set identification a problem with nontrivial consequences if solved. This paper assumes no background knowledge in Markov chains and will introduce all necessary material.

Contents

1 Background

Because our method leverages Markov chains, we introduce the necessary background material. Before jumping straight into Markov chains, we give a contextual primer on how we formulate the transaction set identification problem.

1.1 Framing the problem

Transaction sets can be seen as a sequence of segments. In particular, suppose S is the set of all possible segments. Then a transaction set T can be seen as

$$T = \{s_{i_1}, s_{i_2}, ..., s_{i_n}\}, \text{ where } s_{i_j} \in S.$$

Consider the following example:

```
ISA*00*          *00*          *09*005070479ff    *ZZ*X0000X0        *931001*1020*U*00802*000838602*0*T*^~
GS*AA*21F9W55LN5MAHS*QGE5I*20210310*165655*302208*T*008020~
ST*102*61567539~
ORI*P22MSY1484ZRD7UX6DGUZW4C3Z38~
REF*00*8363c7af-7441-4720-a23b-37a9322c68ec~
BDS*ASB*2*FA~
SE*10*730573370~
GE*10*302208~
IEA*1*000838602~
```

Ignoring the actual content, this EDI document can then be interpreted as a sequence of segments:

$$ISA \to GS \to ST \to ORI \to REF \to BDS \to SE \to GE \to IEA.$$

In this way, an EDI document can be reformulated as a system with multiple "states" such as ISA, BDS, SE, etc. The published standards for each transaction set provide a grammar for the order of these states. This implies that corrupted files are extremely likely to deviate only slightly from the intended transaction set standard. We can therefore adopt a probabilistic perspective:

The transaction set standard with the highest probability should be our best guess.

1.2 Primer on Markov chains

Markov chains describe systems that toggle between various states.[1] In particular, Markov chains make a critical assumption called the *Markov property*: state changes depend only on the current state. For example, consider weather as a Markov chain. Weather switches broadly between three states: sunny, rainy, and snowy. Furthermore, the probability that tomorrow's weather is rainy depends only on **today's** weather - not on yesterday's weather, or any other day before yesterday.

We can encode this probability information into a matrix called the *transition matrix* or *probability matrix*. Continuing with our weather example, suppose the following probabilities:

$$\begin{aligned}
&\mathbb{P}\{\text{sunny} \to \text{sunny}\} = \frac{1}{2}, \mathbb{P}\{\text{sunny} \to \text{rainy}\} = \frac{3}{8}, \mathbb{P}\{\text{sunny} \to \text{snowy}\} = \frac{1}{8}, \\
&\mathbb{P}\{\text{rainy} \to \text{sunny}\} = \frac{1}{4}, \mathbb{P}\{\text{rainy} \to \text{rainy}\} = \frac{1}{2}, \mathbb{P}\{\text{rainy} \to \text{snowy}\} = \frac{1}{4}, \\
&\mathbb{P}\{\text{snowy} \to \text{sunny}\} = \frac{1}{3}, \mathbb{P}\{\text{snowy} \to \text{rainy}\} = \frac{1}{3}, \mathbb{P}\{\text{snowy} \to \text{snowy}\} = \frac{1}{3}.
\end{aligned}$$

[1] Readers who are not interested in the math may skip to the Methods section.

We can then put this information into our transition matrix:

$$\mathbf{P} = \begin{array}{c} \\ \text{sunny} \\ \text{rainy} \\ \text{snowy} \end{array} \begin{array}{c} \begin{array}{ccc} \text{sunny} & \text{rainy} & \text{snowy} \end{array} \\ \begin{pmatrix} \frac{1}{2} & \frac{3}{8} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix} \end{array}$$

Notice that transition matrices define only the transition probabilities from state to state. We cannot know the probability of sunny weather, for example, if we do not have *initial probabilities*. Initial probabilities define the probabilities of starting out at each state. In our weather example, suppose we want to predict the weather starting from today, and a genie has gifted us the actual probabilities of each weather state for today (in order of sunny, rainy, snowy):

$$\pi_0 = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}.$$

Because of the Markov property, we know

$$\begin{aligned} \mathbb{P}\{\text{tomorrow} = \text{sunny}\} &= \sum_{\text{state}} \mathbb{P}\{\text{state} \to \text{sunny}\} \cdot \mathbb{P}\{\text{state}\} \\ &= \frac{1}{2} \cdot \frac{1}{4} + \frac{1}{4} \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{4} = \frac{1}{3}. \end{aligned}$$

In fact, observe that

$$\pi_0 \mathbf{P} = \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 1/2 & 3/8 & 1/8 \\ 1/4 & 1/2 & 1/4 \\ 1/3 & 1/3 & 1/3 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{41}{96} \\ \frac{23}{96} \end{bmatrix}^T = \begin{bmatrix} \mathbb{P}\{\text{tomorrow} = \text{sunny}\} \\ \mathbb{P}\{\text{tomorrow} = \text{rainy}\} \\ \mathbb{P}\{\text{tomorrow} = \text{snowy}\} \end{bmatrix}^T.$$

In general, if π_0 are the initial probabilities, and $\mathbf{P}$ is the transition matrix, then the probabilities of the next step can be computed as $\pi_0 \mathbf{P}$.

2 Method

Given our problem formulation, Markov chains are a natural tool for our identification problem. The strategy is the following:

1. Encode every established transaction set into a transition matrix.
2. Given an EDI document, perform maximum likelihood estimation (MLE) to find the most likely transaction set.

To be precise, consider the following problem formulation. Let T be the set of all established transaction sets, and S be the set of all segments. If there are a total of n segments, define, for each $t \in T$,

$$P(t) \in S^{n \times n} \text{ such that } P(t) \text{ reflects the transition matrix of } t.$$

Then given an EDI document $X = \{s_{i_1}, ..., s_{i_k}\}$, our task is to find

$$
\begin{aligned}
t^* &= \arg\max_{t \in T} \mathbb{P}\{s_{i_1} \to \cdots \to s_{i_k} \mid t\} &(1)\\
&= \arg\max_{t \in T} \mathbb{P}\{s_{i_2} \to \cdots \to s_{i_k} \mid t\} \cdot \mathbb{P}\{s_{i_1} \to s_{i_2} \mid t\} &(2)\\
&= \arg\max_{t \in T} \prod_{j=2}^{k} \mathbb{P}\{s_{i_{j-1}} \to s_j \mid t\} \text{ by the Markov property} &(3)\\
&= \arg\max_{t \in T} \prod_{j=2}^{k} P(t)_{j-1,j}, \text{ where } P(t)_{j-1,j} \text{ is the entry at the } j-1\text{th row and } j\text{th column.} &(4)
\end{aligned}
$$

Finding the most likely transaction set is as simple as iterating through all the possible transaction sets, and identifying which one maximizes equation (4) above.

Our proof of concept encodes Transactions Sets 102, 815, and 993 into transition matrices. When deciding which segments should follow after another, the following were considered:

1. Is the next segment a start/end token (i.e. GS, ST, GE, SE, IEA, ISA)?
2. Is the next segment mandatory (or optional)?
3. Is the next segment allowed to repeat?
4. Is the next segment part of a loop?

For the actual estimation, the log-likelihood (equation (4)) was calculated for every available transaction set, and the transaction set with the highest likelihood was chosen.

The following section describes two encoding methods. The first is the obvious, naive approach; the second leverages a technique from natural language processing (NLP) to improve the robustness of the maximum likelihood estimate.

2.1 Naive approach

If a segment can transition to several segments, then the naive approach is to make all those possible next segments equally likely. For example, if both segments B, C can follow from A, we make

$$\mathbb{P}\{A \to B\} = \mathbb{P}\{A \to C\} = \frac{1}{2}.$$

All the rest of the transition probabilities for that segment are set to 0. Hence, in this example, if segment A cannot transition to D, then $\mathbb{P}\{A \to D\} = 0$.

2.2 k-smoothing approach

The main challenge with the naive approach emerges from assigning zero probability to transitions that are not supposed to happen (as per the published standard.) However, corrupted EDI documents may have incorrect orderings of segments. For example, if the standard asserts that segment B must follow from segment A, a corrupted file may be such that segment A follows B. In this case, the naive method produces $-\infty$ as the log likelihood, wrongly eliminating the correct transaction set from our potential guesses.

To improve the robustness of the naive approach, we leverage a technique from natural language processing called k-smoothing. This method is used when a Markov chain language model encounters a word that is not in its lexicon. Abstractly, k-smoothing works by moving some of the probability of the known lexicon to that of the unknown lexicon, effectively introducing a "cushion" for error.

We use k-smoothing to a similar effect. Instead of setting a zero transition probability, we set it to a small error term. We will walk through an example. Suppose there are 5 possible segments: A, B, C, D, E. The published transaction set dictates that only B, C can follow from A.

We set our smoothing parameter to 1. Then the Ath row of our transition matrix is then as follows:

$$\begin{bmatrix} 1 & 2 & 2 & 1 & 1 \end{bmatrix}.$$

What we have done is add 1 to the states that can follow from A (i.e. B, C). The other states are left untouched. Because every row needs to sum to 1, we normalize the Ath row to the following:

$$\begin{bmatrix} 1 & 2 & 2 & 1 & 1 \end{bmatrix} \to \begin{bmatrix} 1/7 & 2/7 & 2/7 & 1/7 & 1/7 \end{bmatrix}.$$

Using a different smoothing parameter 0.1 yields the following change:

$$\begin{bmatrix} 0.1 & 1.1 & 1.1 & 0.1 & 0.1 \end{bmatrix} \to \begin{bmatrix} 0.04 & 0.44 & 0.44 & 0.04 & 0.04 \end{bmatrix}.$$

We compare the difference between using k-smoothing versus the naive approach. Notice that the naive approach is simply the k-smoothing approach with smoothing parameter 0. With the naive approach on the left and the k-smoothing approach with smoothing parameter 0.1 on the right, we have

$$\begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \end{bmatrix} \text{ vs. } \begin{bmatrix} 0.04 & 0.44 & 0.44 & 0.04 & 0.04 \end{bmatrix}.$$

Notice that the k-smoothing approach allows for some error by shifting some probability to the impossible transitions. With this, our method becomes much more robust to corrupted files that do not obey the standards.

3 Results

15 samples were generated with various corruptions. Some are perfectly legal EDI documents, whereas others are completely illegal (e.g. missing the beginning ISA token, missing other mandatory tokens, etc). The results are summarized in the graph and table below:

Smoothing parameter	0.0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
Accuracy	0.73	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

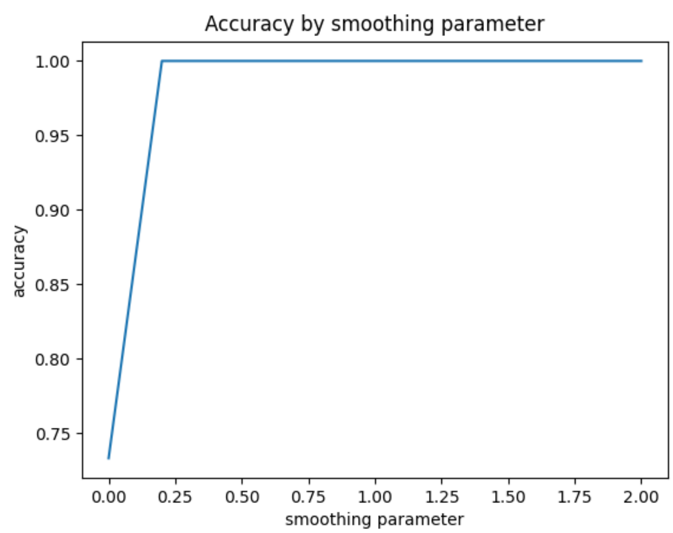

Figure 1: Smoothing parameter μ was varied between 0 and 2, with strong accuracy for $\mu > 0$.

Clearly, more samples must be tested in order to get a better idea of our method's strength. However, for a first proof of concept, our method performs exceptionally well even with properly corrupted files.

4 Discussion

We first explain the advantages of our approach. Our Markov chain / MLE approach is extendable. For instance, the probabilities in the transition matrices can be modified to reflect actual EDI documents. If segment B can *theoretically* follow from segment A but almost never occurs, then the transition probability can be modified to be close to zero.

Not only are the transition probabilities modifiable, but the probabilities of the transaction sets themselves can be added to the model. In other words, we can place a prior on the set of transaction sets T. Placing a prior makes the model much more resistant to overfitting and gives a better estimate of how confident we should be in our guesses. For example, if transaction set 103 occurs much more frequently than transaction set 186 does, their probabilities of occurring can be adjusted accordingly. In enforcing a prior, note that we must adjust our method from an MLE method to a maximum a posteriori (MAP) method. Fortunately, this is not difficult to do; switching from MLEs to MAPs is quickly done, as most machine learning models show.

Next, we expand on some disadvantages, the first of which is changing standards / making new transaction sets. When standards change, *all* the transition probability matrices must be changed also. Furthermore, when a user creates a new transaction set, making a new transition probability matrix for that set may be laborious.

Additionally, computational and storage demands may scale faster than desired. Fortunately, there exist efficient algorithms for MLE estimation (e.g. expectation-maximization), so computational demands are not a major concern. However, because this method demands storing the transition probability matrices for all possible transaction sets, storage could become a potential problem. Every transition probability matrix has size $n \times n$, where n is the number of transaction sets. Hence, storage demands scale quadratically with the number of transaction sets.

Finally, we explore other possible identification methods. Standard machine learning methods can be used. In particular, the following may be fruitful avenues for exploration:

1. *Kneyser-Ney smoothing.* Kneyser-Ney smoothing is another smoothing method that replaces the Markov property. Instead of having the newest state depend only on the current state, Kneyser-Ney smoothing asserts that the newest state depends on other past states also. Parameters are set to control the contribution of each past state to the probability of the newest state; these parameters can also be learned from data.

2. *Support vector machines (supervised).* SVMs are classic models for classification problems, and choosing the correct embedding strategy with SVMs may yield better results.

3. *Positional encodings + multilayer perceptrons (supervised).* As the name suggests, positional encodings encode position/sequence information into the embedding. MLPs can then be used as a standard non-linear classification model.

In general, we recommend *not* using the Markov chain / MLE method, as the disadvantages outweigh the advantages. However, the ideas embedded in Markov chains can be used quite broadly, and should be given further consideration when creating another model.

Appendix

Code

The code can be found in the accompanying zip file.

Results

```
smoothing param: 0.0 | acc: 0.7333333333333333
True label: 102 | Predicted label: 102 | Log prob: -inf
True label: 102 | Predicted label: 102 | Log prob: -0.6931471805599453
True label: 102 | Predicted label: 102 | Log prob: -0.6931471805599453
True label: 102 | Predicted label: 815 | Log prob: -inf
True label: 102 | Predicted label: 815 | Log prob: -inf
True label: 993 | Predicted label: 993 | Log prob: -5.598421958998375
True label: 993 | Predicted label: 993 | Log prob: -3.8066624897703196
True label: 993 | Predicted label: 993 | Log prob: -4.499809670330265
True label: 815 | Predicted label: 815 | Log prob: -3.295836866004329
True label: 993 | Predicted label: 993 | Log prob: -inf
True label: 993 | Predicted label: 993 | Log prob: -4.499809670330265
True label: 815 | Predicted label: 102 | Log prob: -inf
True label: 815 | Predicted label: 815 | Log prob: -3.295836866004329
True label: 815 | Predicted label: 815 | Log prob: -1.0986122886681098
True label: 815 | Predicted label: 102 | Log prob: -inf

smoothing param: 0.2 | acc: 1.0
True label: 102 | Predicted label: 102 | Log prob: -11.781573958709076
True label: 102 | Predicted label: 102 | Log prob: -11.599252401915123
True label: 102 | Predicted label: 102 | Log prob: -13.02636875755527
True label: 102 | Predicted label: 102 | Log prob: -13.391011871143178
True label: 102 | Predicted label: 102 | Log prob: -13.391011871143178
True label: 993 | Predicted label: 993 | Log prob: -16.05068848796117
True label: 993 | Predicted label: 993 | Log prob: -12.677661983265713
True label: 993 | Predicted label: 993 | Log prob: -14.287099895699813
True label: 815 | Predicted label: 815 | Log prob: -13.853463910624953
True label: 993 | Predicted label: 993 | Log prob: -16.415331601549077
True label: 993 | Predicted label: 993 | Log prob: -14.287099895699813
True label: 815 | Predicted label: 815 | Log prob: -13.881634787591649
True label: 815 | Predicted label: 815 | Log prob: -12.426347554984806
True label: 815 | Predicted label: 815 | Log prob: -8.899170370462087
True label: 815 | Predicted label: 815 | Log prob: -13.881634787591649

smoothing param: 0.4 | acc: 1.0
True label: 102 | Predicted label: 102 | Log prob: -14.278029353500413
True label: 102 | Predicted label: 102 | Log prob: -14.991379241377878
True label: 102 | Predicted label: 102 | Log prob: -16.852131582092884
True label: 102 | Predicted label: 102 | Log prob: -16.244142209873246
True label: 102 | Predicted label: 102 | Log prob: -16.244142209873246
True label: 993 | Predicted label: 993 | Log prob: -19.68262078931966
True label: 993 | Predicted label: 993 | Log prob: -15.655084896769669
True label: 993 | Predicted label: 993 | Log prob: -17.621197753142503
True label: 815 | Predicted label: 815 | Log prob: -17.34878315282151
True label: 993 | Predicted label: 993 | Log prob: -19.074631417100022
```

```
    True label: 993 | Predicted label: 993 | Log prob: -17.621197753142503
    True label: 815 | Predicted label: 815 | Log prob: -16.540123085139722
    True label: 815 | Predicted label: 815 | Log prob: -15.488030812106505
    True label: 815 | Predicted label: 815 | Log prob: -11.36518473975219
    True label: 815 | Predicted label: 815 | Log prob: -16.540123085139722

    smoothing param: 0.6 | acc: 1.0
    True label: 102 | Predicted label: 102 | Log prob: -15.645449350522336
    True label: 102 | Predicted label: 102 | Log prob: -16.83367379788013
    True label: 102 | Predicted label: 102 | Log prob: -18.928619526095932
    True label: 102 | Predicted label: 102 | Log prob: -17.814503050891858
    True label: 102 | Predicted label: 102 | Log prob: -17.814503050891854
    True label: 993 | Predicted label: 993 | Log prob: -21.721131771828432
    True label: 993 | Predicted label: 993 | Log prob: -17.314031499602432
    True label: 993 | Predicted label: 993 | Log prob: -19.483085199971956
    True label: 815 | Predicted label: 815 | Log prob: -19.28381408486423
    True label: 993 | Predicted label: 993 | Log prob: -20.607015296624358
    True label: 993 | Predicted label: 993 | Log prob: -19.483085199971956
    True label: 815 | Predicted label: 815 | Log prob: -18.02659676601948
    True label: 815 | Predicted label: 815 | Log prob: -17.188868356648427
    True label: 815 | Predicted label: 815 | Log prob: -12.71277521293548
    True label: 815 | Predicted label: 815 | Log prob: -18.02659676601948

    smoothing param: 0.8 | acc: 1.0
    True label: 102 | Predicted label: 102 | Log prob: -16.528916970295008
    True label: 102 | Predicted label: 102 | Log prob: -18.020571847072723
    True label: 102 | Predicted label: 102 | Log prob: -20.26599852622682
    True label: 102 | Predicted label: 102 | Log prob: -18.831502063289054
    True label: 102 | Predicted label: 102 | Log prob: -18.831502063289054
    True label: 993 | Predicted label: 993 | Log prob: -23.056411204378797
    True label: 993 | Predicted label: 993 | Log prob: -18.397173797120434
    True label: 993 | Predicted label: 993 | Log prob: -20.69975889011448
    True label: 815 | Predicted label: 815 | Log prob: -20.542517017717547
    True label: 993 | Predicted label: 993 | Log prob: -21.62191474144103
    True label: 993 | Predicted label: 993 | Log prob: -20.69975889011448
    True label: 815 | Predicted label: 815 | Log prob: -18.996794919669554
    True label: 815 | Predicted label: 815 | Log prob: -18.29709033856345
    True label: 815 | Predicted label: 815 | Log prob: -13.583785710034807
    True label: 815 | Predicted label: 815 | Log prob: -18.996794919669554

    smoothing param: 1.0 | acc: 1.0
    True label: 102 | Predicted label: 102 | Log prob: -17.15277398070429
    True label: 102 | Predicted label: 102 | Log prob: -18.857522072942714
    True label: 102 | Predicted label: 102 | Log prob: -21.208897330106193
    True label: 102 | Predicted label: 102 | Log prob: -19.55066925350266
    True label: 102 | Predicted label: 102 | Log prob: -19.55066925350266
    True label: 993 | Predicted label: 993 | Log prob: -24.00754130903163
    True label: 993 | Predicted label: 993 | Log prob: -19.167299000864055
    True label: 993 | Predicted label: 993 | Log prob: -21.565194273662428
    True label: 815 | Predicted label: 815 | Log prob: -21.43529264908848
    True label: 993 | Predicted label: 993 | Log prob: -22.349313232428095
    True label: 993 | Predicted label: 993 | Log prob: -21.565194273662428
    True label: 815 | Predicted label: 815 | Log prob: -19.686092794279222
```

```
    True label: 815 | Predicted label: 815 | Log prob: -19.083917391925002
    True label: 815 | Predicted label: 815 | Log prob: -14.199223321186592
    True label: 815 | Predicted label: 815 | Log prob: -19.686092794279222

    smoothing param: 1.2 | acc: 1.0
    True label: 102 | Predicted label: 102 | Log prob: -17.619065055097828
    True label: 102 | Predicted label: 102 | Log prob: -19.482568429184724
    True label: 102 | Predicted label: 102 | Log prob: -21.912986893688654
    True label: 102 | Predicted label: 102 | Log prob: -20.08870423275504
    True label: 102 | Predicted label: 102 | Log prob: -20.088704232755042
    True label: 993 | Predicted label: 993 | Log prob: -24.722708486719245
    True label: 993 | Predicted label: 993 | Log prob: -19.745689803421975
    True label: 993 | Predicted label: 993 | Log prob: -22.215328981079185
    True label: 815 | Predicted label: 815 | Log prob: -22.10464930394376
    True label: 993 | Predicted label: 993 | Log prob: -22.89842582578563
    True label: 993 | Predicted label: 993 | Log prob: -22.215328981079185
    True label: 815 | Predicted label: 815 | Log prob: -20.203405601874017
    True label: 815 | Predicted label: 815 | Log prob: -19.67423083943983
    True label: 815 | Predicted label: 815 | Log prob: -14.65947182815971
    True label: 815 | Predicted label: 815 | Log prob: -20.203405601874014

    smoothing param: 1.4 | acc: 1.0
    True label: 102 | Predicted label: 102 | Log prob: -17.981786149160705
    True label: 102 | Predicted label: 102 | Log prob: -19.96851829273627
    True label: 102 | Predicted label: 102 | Log prob: -22.460345385368843
    True label: 102 | Predicted label: 102 | Log prob: -20.50751479346896
    True label: 102 | Predicted label: 102 | Log prob: -20.507514793468957
    True label: 993 | Predicted label: 993 | Log prob: -25.281458332977444
    True label: 993 | Predicted label: 993 | Log prob: -20.19721122153794
    True label: 993 | Predicted label: 993 | Log prob: -22.722939865846197
    True label: 815 | Predicted label: 815 | Log prob: -22.62651795718918
    True label: 993 | Predicted label: 993 | Log prob: -23.328627741077558
    True label: 993 | Predicted label: 993 | Log prob: -22.722939865846197
    True label: 815 | Predicted label: 815 | Log prob: -20.60699599079062
    True label: 815 | Predicted label: 815 | Log prob: -20.134690864556607
    True label: 815 | Predicted label: 815 | Log prob: -15.017653930294117
    True label: 815 | Predicted label: 815 | Log prob: -20.60699599079062

    smoothing param: 1.6 | acc: 1.0
    True label: 102 | Predicted label: 102 | Log prob: -18.272480630855007
    True label: 102 | Predicted label: 102 | Log prob: -20.35782189466203
    True label: 102 | Predicted label: 102 | Log prob: -22.898818011101074
    True label: 102 | Predicted label: 102 | Log prob: -20.84332971044373
    True label: 102 | Predicted label: 102 | Log prob: -20.843329710443733
    True label: 993 | Predicted label: 993 | Log prob: -25.730745978786665
    True label: 993 | Predicted label: 993 | Log prob: -20.56006028273596
    True label: 993 | Predicted label: 993 | Log prob: -23.130909362324687
    True label: 815 | Predicted label: 815 | Log prob: -23.045486548020197
    True label: 993 | Predicted label: 993 | Log prob: -23.675257678129324
    True label: 993 | Predicted label: 993 | Log prob: -23.130909362324687
    True label: 815 | Predicted label: 815 | Log prob: -20.93115774733992
    True label: 815 | Predicted label: 815 | Log prob: -20.50449043158115
    True label: 815 | Predicted label: 815 | Log prob: -15.304817198657197
```

```
    True label: 815 | Predicted label: 815 | Log prob: -20.931157747339917

    smoothing param: 1.8 | acc: 1.0
    True label: 102 | Predicted label: 102 | Log prob: -18.510922220520502
    True label: 102 | Predicted label: 102 | Log prob: -20.67705621078669
    True label: 102 | Predicted label: 102 | Log prob: -23.258354706249758
    True label: 102 | Predicted label: 102 | Log prob: -21.118888963065732
    True label: 102 | Predicted label: 102 | Log prob: -21.11888896306573
    True label: 993 | Predicted label: 993 | Log prob: -26.100238556229172
    True label: 993 | Predicted label: 993 | Log prob: -20.858329584735454
    True label: 993 | Predicted label: 993 | Log prob: -23.466296327280684
    True label: 815 | Predicted label: 815 | Log prob: -23.38961765962386
    True label: 993 | Predicted label: 993 | Log prob: -23.960772813045146
    True label: 993 | Predicted label: 993 | Log prob: -23.466296327280684
    True label: 815 | Predicted label: 815 | Log prob: -21.19750818295441
    True label: 815 | Predicted label: 815 | Log prob: -20.808319164160793
    True label: 815 | Predicted label: 815 | Log prob: -15.540434706263818
    True label: 815 | Predicted label: 815 | Log prob: -21.19750818295441

    smoothing param: 2.0 | acc: 1.0
    True label: 102 | Predicted label: 102 | Log prob: -18.710183554361553
    True label: 102 | Predicted label: 102 | Log prob: -20.943775775868648
    True label: 102 | Predicted label: 102 | Log prob: -23.558735553904846
    True label: 102 | Predicted label: 102 | Log prob: -21.34924088397681
    True label: 102 | Predicted label: 102 | Log prob: -21.34924088397681
    True label: 993 | Predicted label: 993 | Log prob: -26.409669901974823
    True label: 993 | Predicted label: 993 | Log prob: -21.10802474533411
    True label: 993 | Predicted label: 993 | Log prob: -23.74708207494937
    True label: 815 | Predicted label: 815 | Log prob: -23.677522149293548
    True label: 993 | Predicted label: 993 | Log prob: -24.20017523204679
    True label: 993 | Predicted label: 993 | Log prob: -23.74708207494937
    True label: 815 | Predicted label: 815 | Log prob: -21.420399430376257
    True label: 815 | Predicted label: 815 | Log prob: -21.06256237125735
    True label: 815 | Predicted label: 815 | Log prob: -15.737386717206444
    True label: 815 | Predicted label: 815 | Log prob: -21.420399430376257

```