9. n-gram-language models

9.1 Statistical modeling

1. Take some data (generated from unknown probability distribution)
2. Make an estimate of the distribution based on data
3. Make inferences based on the estimate

Tasks of the modeling:

- Dividing data into equivalence classes
- Finding a good statistical estimator for each class
- Combining several estimators

Typical assumption: **stationarity**: probability distribution of the data does not change essentially over time.

Why statistical language modeling

Classical task: prediction of next word (or letter) based on words (or letters) already seen ('Shannon game'). Can be used, for instance, in the following applications:

- speech recognition
- OCR
- statistical MT

Estimation methods are common, can be used also in other tasks (e.g. word sense disambiguation, parsing)

9.2 N-gram models

n-gram model: one predicts word w_n based on previous $n-1$ words:

$$P(w_n|w_1 w_2 \cdots w_{n-1}) \tag{1}$$

Equation can also be given as $P(w_t|w_{t-(n-1)} w_{t-(n-2)} \cdots w_{t-1})$ where t is the index of the word in the whole material.

Example: the slides of this lecture (in Finnish), n=4:

		w_{t-3}	w_{t-2}	w_{t-1}	w_t		
...	sitä	enemmän	dataa	tarvitaan	**mallin**	estimointiin	...

Names for models

n=1	unigram
n=2	bigram
n=3	trigram
n=4	4-gram, fourgram

Connection to equivalence classes: in n-gram models every $n-1$ word history

gets its own equivalence class.

Same n-gram property from another point of view: model assumes that a word depends only on the previous $(n-1)$ words but not from any further than thata (Markov assumption).

Markov model: k'th degree Markov model is a model that puts all histories of the length k:n into the same equivalence class. In other words, n-gram model is a Markov model of degree $n-1$.

Examples:
Sue swallowed the large green ____

Samppa Lajunen voitti kultaa ____

Growth of the number of parameters

	Model	Parameters if vocab 20,000
n=1	unigram	20000
n=2	bigram	$20000^2 = 400$ million
n=3	trigram	$20000^3 = 8$ billion
n=4	4-gram, fourgram	1.6×10^{17}

9.3 Dividing features into equivalnce classes

- Features (continuous and discrete) can be divided into equivalence class bins.
- E.g.. continuous variable 'age' divided into classes 0-2; 3-5; 7-10; 11-15; 16-25; 26-35 etc.
- The higher the number of eq. classes the more data is needed for model estimation
- On the other hand, if the number of classes is low the value of the variable cannot be predicted accurately.

Example: Prediction based on:
1. three previous part of speech tags (noun, verb, adj, num etc.) OR
2. three previous words

1. less data and less accurate estimates
2. more accurate estimates but much more needed

Some ways to form equivalence classes

- 'Features are' -¿ 'features are'
- Tranforming inflected word forms into the basic word form ('saunan', 'saunalle', saunalta', 'saunojemme', etc. -¿ 'sauna')
- Grouping based on POS tag (those words that have same syntactic role form an eq. class)
- Grouping based on semantic information (those words that have the same or similar meaning form an eq. class)

It is beneficial that the words would "behave" in a similar manner within an eq. class.

Different ways to take the history into account

- Some features are selected from the history without considering their position in time, e.g. model: $P(w_t|\text{predicate of the sentence}, w_{t-1})$
- Instead of stream of words, a bag of words is considered, i.e. not taking into account word order:
  $P(w_n|w_1, w_2, \ldots, w_{n-1})$

9.4 n-gram model statistical estimation

Gioven: set of sample from each equivalence class (bin) From Bayesin rule:

$$P(w_n|w_1 \cdots w_{n-1}) = \frac{P(w_1 \cdots w_n)}{P(w_1 \cdots w_{n-1})} \tag{2}$$

Modeln optimization: maximize datan probability (i.e. product of word probabilities).

Notation:

N	Number of training samples
B	Number of eq. classes (bins)
w_{1n}	n-gram $w_1 \cdots w_n$
$C(w_1 \cdots w_n)$	n-gram $w_1 \cdots w_n$ number in training data
r	number of n-grams
N_r	Number of those bins in which there are r samples
h	history (preceeeding words)

Maximum likelihood-estimate (MLE)

$$P_{\mathsf{MLE}}(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n)}{N} \tag{3}$$

$$P_{\mathsf{MLE}}(w_n | w_1 \cdots w_{n-1}) = \frac{C(w_1 \cdots w_n)}{C(w_1 \cdots w_{n-1})} \tag{4}$$

- MLE-estimation leads into such parameter selection that the probability of training data maksimized.
  (within inpendence assumptions)
- The whole prob. mass is divided among the cases that appear in the training data, following ratios of frequency.
- Thus, gives P=0 for a case that does not appear in the training data
- Because the overall P is calculated as a product of the individual word probabilities, any zero value will cause the overall P to be zero.

- Example of data sparseness: within the first 1.5 million words (IBM laser patent text corpus) 23% subsequent trigrams were previously unseen.
- MLE is thus not a very useful estimatefor sparse data, such as n-grams.
- One needs a systematic approach to take into account previously unseen word and n-gram probabilities. This is called *smoothing*

Table 6.3: MLE-estimates from Austen's books for some n-grams of a sentence

In person	*she*		*was*		*inferior*		*to*	
1-gram	$P(\cdot)$		$P(\cdot)$		$P(\cdot)$		$P(\cdot)$	
1	the	0.034	the	0.034	the	0.034	the	0.034
2	to	0.032	to	0.032	to	0.032	**to**	**0.032**
3	and	0.030	and	0.030	and	0.030		
4	of	0.029	of	0.029	of	0.029		
...								
8	was	0.015	**was**	**0.015**	was	0.015		
...								
13	**she**	**0.011**			she	0.011		
...								
254					both	0.0005		
...								
435					sisters	0.0003		
...								
1701					**inferior**	**0.00005**		
2-gram	$P(\cdot \mid person)$		$P(\cdot \mid she)$		$P(\cdot \mid was)$		$P(\cdot \mid inferior)$	
1	and	0.099	had	0.141	not	0.065	**to**	**0.212**
2	who	0.099	**was**	**0.122**	a	0.052		
3	to	0.076			the	0.033		
4	in	0.045			to	0.031		
...								
23	**she**	**0.009**						
...								

Laplace law

Some probability is "moved" to unseen cases by adding one to each frequency count:

$$P_{\mathsf{LAP}}(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n) + 1}{N + B} \tag{5}$$

- This corresponds to the Bayes estimate with the prior that all cases are equally probable.
- If the data is very sparse, this gives too much probability mass to the previously unseen cases.
- One could ask should one add, for instance, 0.01 or 0.001 rather than 1?

Estimates of expected frequency:

$r = f_{MLE}$	$f_{empirical}$	f_{Lap}	f_{del}	f_{GT}	N_r	T_r
0	0.000027	0.000137	0.000037	0.000027	74 671 100 000	2 019 187
1	0.448	0.000274	0.396	0.446	2 018 046	903 206
2	1.25	0.000411	1.24	1.26	449 721	564 153
3	2.24	0.000548	2.23	2.24	188 933	424 015
4	3.23	0.000685	3.22	3.24	105 668	341 099
5	4.21	0.000822	4.22	4.22	68 379	287 776
6	5.23	0.000959	5.20	5.19	48 190	251 951
7	6.21	0.00109	6.21	6.21	35 709	221 693
8	7.21	0.00123	7.18	7.24	27 710	199 779
9	8.26	0.00137	8.18	8.25	22 280	183 971

Table 6.4 Estimated frequencies for the AP data from Church and Gale (1991a). The first five columns show the estimated frequency calculated for a bigram that actually appeared r times in the training data according to different estimators: r is the maximum likelihood estimate, $f_{empirical}$ uses validation on the test set, f_{Lap} is the 'add one' method, f_{del} is deleted interpolation (two-way cross validation, using the training data), and f_{GT} is the Good-Turing estimate. The last two columns give the frequencies of frequencies and how often bigrams of a certain frequency occurred in further text.

Lidstone law, Jeffreys-Perks law

$$P_{\text{Lid}}(w_1 \cdots w_n) = \frac{C(w_1 \cdots w_n) + \lambda}{N + B\lambda} \tag{6}$$

This corresponds to linear interpolation between the prior of uniform distribution and MLE estimate. Let's set $\mu = N/(N + B\lambda)$:

$$P_{\text{Lid}}(w_1 \cdots w_n) = \mu \frac{C(w_1 \cdots w_n)}{N} + (1 - \mu)\frac{1}{B} \tag{7}$$

- Jeffreys prior: $\lambda = 1/2$, i.e. add $1/2$ to each frequency lukumäärään $1/2$. This is also called *Expected Likelihood Estimation* (ELE)
- One has to choose the value of λ in one way or another
- With low frequencies this does not coincide well with the real distribution either

Good-Turing estimator

See frequence histograms in Table 6.7.

$$P_{GT}(w_1 \cdots w_n) = \frac{r^*}{N}, \text{ jossa } r^* = \frac{(r+1)S(r+1)}{S(r)} \tag{8}$$

$S(r)$ is expected value for N_r

Simple Good-Turing estimator: choose a power function: $S(r) = ar^b$ in which the parameters a and b are fitted with the frequences according to the frequency histogram.

This is a quite good estimator that is commonly in use.

	Bigrams				Trigrams		
r	N_r	r	N_r	r	N_r	r	N_r
1	138741	28	90	1	404211	28	35
2	25413	29	120	2	32514	29	32
3	10531	30	86	3	10056	30	25
4	5997	31	98	4	4780	31	18
5	3565	32	99	5	2491	32	19
6	2486		...	6	1571		...
7	1754	1264	1	7	1088	189	1
8	1342	1366	1	8	749	202	1
9	1106	1917	1	9	582	214	1
10	896	2233	1	10	432	366	1
	...	2507	1		...	378	1

Table 6.7 Extracts from the frequencies of frequencies distribution for bigrams and trigrams in the Austen corpus.

Other smoothing methods

Term 'discounting' refers to the idea that the P of seen n-grams is moved to the unseen n-grams.

- Absolute discounting: From all seen n-grams a constant probability mass σ is removed and divided evenly among unseen n-grams.
- Linear discounting: The probabilities of seen n-grams is scaled with a constant that is smaller than 1, and the remaining probability mass is divided evenly among unseen ones. This approach is not particularly good as it "punishes" the frequent n-grams relatively mor (the estimates of which are, however, better).
- Witten-Bell discounting: The probability mass of surprising events is estimated based on the fact how common it has been so far to encounter unseen events: $\sum_{i:C(i)=0} p_i = \frac{T}{N+T}$ where T is the number of bins seen so far

These methods differ from each other in what kind of assumptions are made concerning cases that have not been seen and their relation to the cases that have been seen.

E.g. CMU Statistical Language Toolkit implements several different discounting ja smoothing methods for n-grams.

9.5 Combining estimators

- Previously we have considered a situation in which one tries to estimate an identical probability for all n-grams of particular size that have been seen.
- However, if the parts of an n-gram are frequent, should one not use that information in estimating the probability of the whole n-gram?
- The basic motivation is the smoothing of the estimate of more generally combining different information sources.

Linear interpolation

(also: mixture models tai sum of experts)

A weighted average is calculated based on the estimates of contexts of different lengths: estimaateista:

$$P_{li}(w_n|w_{n-2}w_{n-1}) = \lambda_1 P_1(w_n) + \lambda_2 P_2(w_n|w_{n-1}) + \lambda_3 P_3(w_n|w_{n-2}w_{n-1}) \quad (9)$$

($0 \leq \lambda_1 \leq 1$ ja $\sum_i \lambda_1 = 1$)

Parameters λ can be set manually or optimized with data.

General linear interpolation

Previously parameters λ did not depend on the words, and thus the parameter is constant for all n-grams of particular size.

In a more general approach, the parameters can be set to depend on the

history:

$$P_{li}(w|h) = \sum_i \lambda_i(h) P_i(w|h) \tag{10}$$

($0 \leq \lambda_1 \leq 1$ ja $\sum_i \lambda_1 = 1$) and optmize them using, for instance, EM algorithm. However, if there is an own λ for each history, the sparseness of data problem is again encountered, and some smoothing is needed, like the equivalence classes of the history, etc.

Backing off

- Principle: look for the most specific model that gives "sufficiently enough" information on the data in the current context
- In the other words, one backs off from using long contexts to shorter ones. One can decide to "believe" an estimate if it is based on at least k samples (k may be e.g. 1 or 2)
- Criticism: Adding new training data may influence a lot the probabilities when it causes changes for many different words concerning the n-gram frequencies related to them.
- However, the models are simple and work reasonably well and therefore they are commonly in use.
- Back-off model is a special case of general lineara interpolation: $\lambda_i(h) = 1$ when k's value is large enough, 0 otherwise.
- This approach resembles Dynamically Expanding Context (DEC) method by Kohonen. Consider also VariKN Language Modeling toolkit by Siivola.

Back-off- model usage example:

	$P(she\,h)$	$P(was\|h)$	$P(inferior\|h)$	$P(to\|h)$	$P(both\|h)$	$P(sisters\|h)$
Unigram	0.011	0.015	0.00005	0.032	0.0005	0.0003
Bigram	0.00529	0.1219	0.0000159	0.183	0.000449	0.00372
n used	2	2	1	2	2	2
Trigram	0.00529	0.0741	0.0000162	0.183	0.000384	0.00323
n used	2	3	1	2	2	2

Table 6.11 Probability estimates of the test clause according t
guage models. The unigram estimate is our previous MLE unigram
other two estimates are back-off language models. The last col
overall probability estimate given to the clause by the model.

9.6 Model estimation in general

This applies to any model comparison, not only n-grams or language models.

Held-out estimation

Usually data is divided into three parts before method development / model estimation:

- **Training set:** data with which the model is trained
- **Validation set:** data independent from training set with which parameters are selected (e.g. previously mentioned λ)
- **Test set:** randomly selected data independent from training and validation set with which the model is evaluated. (for instance 10% of the training data),

Test set has to be kept completely separate from other parts of the data during method development. If the test data is allowed to influence during

the method development, it is not suitable for testing in the end anymore.

However, method development is often a cyclical process in which methods are refined and results tested over and over again. Therefore one may have a division between:
1. **Development test set** that is used to compare the variants of the method being developed
2. **Final test set** that is used to produce final published results that have not been used for anything before

There are different options for choosing test set (and validation set):
1. random selections (random short text fragments) 2. longer portions of the corpus (for instance, later parts of the data)

The second approach matches usually better real usage situations. It also gives more realistic, usually slightly worse results because rarely phenomena are fully stationary (for instance in news corpora new names are being introduced over time, etc.)

Comparison of methods

If one compares just averages, it is not possible to know whether the differences between results significant.

One solution: The variance of the results is also measured for different data sets and the statistical significance is measured using, e.g., the *t test*.

	System 1	System 2
scores	71, 61, 55, 60, 68, 49, 42, 72, 76, 55, 64	42, 55, 75, 45, 54, 51 55, 36, 58, 55, 67
total	609	526
n	11	11
mean $\bar{x}_i$	55.4	47.8
$s_i^2 = \sum(x_{ij} - \bar{x}_i)^2$	1,375.4	1,228.8
df	10	10

$$\text{Pooled } s^2 = \frac{1375.4+1228.8}{10+10} \approx 130.2$$

$$t = \frac{\bar{x}_1-\bar{x}_2}{\sqrt{\frac{2s^2}{n}}} = \frac{55.4-47.8}{\sqrt{\frac{2\cdot 130.2}{11}}} \approx 1.56$$

Table 6.6 Using the t test for comparing the performance of two systems. Since we calculate the mean for each data set, the denominator in the calculation of variance and the number of degrees of freedom is $(11 - 1) + (11 - 1) = 20$. The data do not provide clear support for the superiority of system 1. Despite the clear difference in mean scores, the sample variance is too high to draw any definitive conclusions.

Cross validation

- Data is divided into K parts among which one at the time is test data, others training data. is repeated in such a way that each set is training data in its turn. K is a number between 2. . . N, where N is the number of samples.
- Advantage: All data influences both model training and testing. Thus the data is used maximally which is important expecially when there is limited amount of data available.
- There are several different variants (deleted estimation, leave-one-out-estimation)

Both cross validation and held-out-estimation can be used to choose model parameters, and thus, for instance, smoothen the probability estimates.

9.7 n-gram model critique

problems of n-grams as language models:

- Neglecting long distance dependencies
- Neglecting word order
- Need for smoothing shows that there is a structural problem
- Dependencies are estimated directly between words. It would be better that the dependencies would be modeled between some latent variables.
- However: n-gram model combines syntactic and semantic short context dependecies in a rather well functioning manner, at least for English
- Model optimization and improving smoothing methods has been an active topic of research. It is possible that a vertain optimum has been reached within the family of models.