Probabilistic Machine Learning

Lecture 19

Example: Topic Models

Philipp Hennig

28 June 2021

Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning
<table>
<thead>
<tr>
<th>#</th>
<th>date</th>
<th>content</th>
<th>Ex</th>
<th>#</th>
<th>date</th>
<th>content</th>
<th>Ex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20.04.</td>
<td>Introduction</td>
<td></td>
<td>14</td>
<td>09.06.</td>
<td>Generalized Linear Models</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>21.04.</td>
<td>Reasoning under Uncertainty</td>
<td></td>
<td>15</td>
<td>15.06.</td>
<td>Exponential Families</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>27.04.</td>
<td>Continuous Variables</td>
<td>2</td>
<td>16</td>
<td>16.06.</td>
<td>Graphical Models</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>28.04.</td>
<td>Monte Carlo</td>
<td></td>
<td>17</td>
<td>22.06.</td>
<td>Factor Graphs</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>04.05.</td>
<td>Markov Chain Monte Carlo</td>
<td>3</td>
<td>18</td>
<td>23.06.</td>
<td>The Sum-Product Algorithm</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>05.05.</td>
<td>Gaussian Distributions</td>
<td></td>
<td>19</td>
<td>29.06.</td>
<td>Example: Modelling Topics</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>11.05.</td>
<td>Parametric Regression</td>
<td>4</td>
<td>20</td>
<td>30.06.</td>
<td>Mixture Models</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>12.05.</td>
<td>Learning Representations</td>
<td></td>
<td>21</td>
<td>06.07.</td>
<td>EM</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>18.05.</td>
<td>Gaussian Processes</td>
<td>5</td>
<td>22</td>
<td>07.07.</td>
<td>Variational Inference</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>19.05.</td>
<td>Understanding Kernels</td>
<td></td>
<td>23</td>
<td>13.07.</td>
<td>Fast Variational Inference</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>25.05.</td>
<td>An Example for GP Regression</td>
<td>6</td>
<td>25</td>
<td>20.07.</td>
<td>Outlook</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>08.06.</td>
<td>GP Classification</td>
<td>7</td>
<td>26</td>
<td>21.07.</td>
<td>Revision</td>
<td></td>
</tr>
</tbody>
</table>
The Toolbox

Framework:

\[ \int p(x_1, x_2) \, dx_2 = p(x_1) \quad p(x_1, x_2) = p(x_1 \mid x_2)p(x_2) \quad p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} \]

Modelling:
- graphical models
- Gaussian distributions
- (deep) learnt representations
- Kernels
- Markov Chains
- Exponential Families / Conjugate Priors
- Factor Graphs & Message Passing

Computation:
- Monte Carlo
- Linear algebra / Gaussian inference
- maximum likelihood / MAP
- Laplace approximations
the goal for (most of) the rest of the course:
Build a Model of History
[The President] shall from time to time give to the Congress Information of the State of the Union, and recommend to their Consideration such Measures as he shall judge necessary and expedient.

Article II, §3 of the US Constitution

- Delivered annually since 1790
- Summarizes affairs of the US federal government
- Historically delivered in writing, generally spoken since 1982,
- On radio since 1923, TV since 1947, in the evenings since 1965, webcast since 2002
- The inaugural SotU of a new president typically has a different tone
[The President] shall from time to time give to the Congress Information of the State of the Union, and recommend to their Consideration such Measures as he shall judge necessary and expedient.

Article II, §3 of the US Constitution

Delivered annually since 1790

Summarizes affairs of the US federal government

historically delivered in writing, generally spoken since 1982,

on radio since 1923, TV since 1947, in the evenings since 1965, webcast since 2002

the inaugural SotU of a new president typically has a different tone
The SotU Addresses are not a perfect reflection of US history, but they are ...

- available in their entirety online
- available without interruption for over 200 years
- topical
- given in a reasonably similar setting, annually

Our task: Find topics of US history over time.

This is an unsupervised dimensionality reduction task.
Disclaimer:

► This is not a course in natural language processing!
► There is an entire toolbox of models for text analysis that will not be discussed here. Some of them have probabilistic interpretation, others don’t.
► The point of this exercise is to try out the tools developed in this course on a practical problem. There is no claim that this is the “best” thing to do.

However, the model ultimately developed here is likely unusually expressive in its structure, and more flexible than the standard tools. Key takeaway: It does pay to spend time developing your model!

Our Goal: Build *craftware*: customized, effective and efficient solution to the learning task. Use toolboxes where they help, be willing to write our own solution where necessary.
A Look at the Data

D = 231 documents (1790 – 2019; 2 in 1961 (Eisenhower & JFK))

individual documents of length $I_d \sim 10^3$ words

$V \sim 10\,000$ words in vocabulary

A few first simplifications

- there are many redundant stop words required for human understanding but carrying only negligible semantic information
- since we are looking to reduce complexity, we necessarily have to throw out a bit of structure
- e.g., usage of word is significant, but its position in the text is not crucial. We will model the texts as Bags of Words
Bags of Words
A look at the data
A Reduced Representation

low-rank decomposition

\[ X \sim U^T Q \]

Probabilistic ML — P. Hennig, SS 2021 — Lecture 19: The Sum-Product Algorithm — © Philipp Hennig, 2021 CC BY-NC-SA 3.0
Consider a dataset $X \in \mathbb{R}^{D \times V}$. **Dimensionality Reduction** aims to find an encoding $\phi : \mathbb{R}^{V} \to \mathbb{R}^{K}$ and a decoding $\psi : \mathbb{R}^{K} \to \mathbb{R}^{V}$ with $K \ll V$ such that the encoded representation

$$Z := \phi(X) \in \mathbb{R}^{D \times K}$$

is a **good approximation** of $X$ in the sense that some **reconstruction loss** of $\tilde{X} = \psi(Z)$,

$$\mathcal{L}(X, \psi(Z)) = \mathcal{L}(X, \psi \circ \phi(X))$$

is minimized or small. This may be done, e.g., to

- save memory
- construct a low-dimensional visualization
- “find structure”
Linear dimensionality reduction

The classic derivation of PCA

Data: $X \in \mathbb{R}^{D \times V} = [x_1; \ldots; x_D]$.

Consider an orthonormal basis $\{u_i\}_{i=1,\ldots,V}$, $u_i^T u_j = \delta_{ij}$. Then

$$x_d = \sum_{i=1}^{V} (x_d^T u_i) u_i = \sum_{i=1}^{V} \alpha_i u_i \quad X = (XU)U^T$$

An approximation in $K < D$ degrees of freedom is given by any set $(A, b, U)$ as

$$\tilde{x}_d := \sum_{k=1}^{K} a_k u_k + \sum_{\ell=K+1}^{V} b_{\ell} u_{\ell}$$

What is the best approximation?
Let’s find \((A, b, U)\) to minimize the square empirical risk

\[
J = \frac{1}{D} \sum_{d=1}^{D} \|x_d - \tilde{x}_d\|^2 = \frac{1}{D} \sum_{d=1}^{D} \sum_{v=1}^{V} \left[ x_d - \sum_{k=1}^{K} a_{dk} u_k - \sum_{j=K+1}^{V} b_j u_j \right]_v^2
\]

First, let’s find \(a_{dk}\) and \(b_j\): Recall \(\sum_j u_{ij} u_{kj} = \delta_{ik}\), use \(\bar{x} := \frac{1}{D} \sum_d x_d\), to find

\[
\frac{\partial J}{\partial a_{d\ell}} = \frac{2}{D} \sum_{v=1}^{V} \left[ x_d - \sum_{k=1}^{K} a_{dk} u_k - \sum_{j=K+1}^{V} b_j u_j \right]_v (-u_{\ell v}) = \frac{2}{D} (-x_d^T u_{\ell}) + \frac{2}{D} a_{d\ell} \overset{!}{=} 0
\]

\[
\frac{\partial J}{\partial b_{\ell}} = \frac{2}{D} \sum_{d=1}^{D} \sum_{v=1}^{V} \left[ x_d - \sum_{k=1}^{K} a_{dk} u_k - \sum_{j=K+1}^{V} b_j u_j \right]_v (-u_{\ell v}) = \frac{2}{D} \sum_{d=1}^{D} (-x_d^T u_{\ell}) + 2b_{\ell} \overset{!}{=} 0
\]

Thus \(a_{dk} = x_d^T u_k\), and \(b_j = \bar{x}^T u_j\).
The best approximation

Empirical Risk Minimization derivation of PCA

With $a_{dk} = x_d^T u_k$, $b_j = \bar{x}^T u_j$, things simplify:

$$x_d - \tilde{x}_d = x_d - \sum_{k=1}^{K} a_{dk} u_k - \sum_{j=K+1}^{V} b_j u_j = \sum_{\ell=1}^{V} (x_d^T u_{\ell}) u_{\ell} - \sum_{k=1}^{K} (x_d^T u_k) u_k - \sum_{j=K+1}^{V} (\bar{x}^T u_j) u_j$$

$$= \sum_{\ell=1}^{V} (x_d^T u_{\ell}) u_{\ell} - \sum_{k=1}^{K} (x_d^T u_k) u_k + \sum_{\ell=K+1}^{V} (x_d^T u_{\ell}) u_{\ell} - \sum_{j=K+1}^{V} (\bar{x}^T u_j) u_j$$

$$= \sum_{j=K+1}^{V} ((x_d - \bar{x})^T u_j) u_j$$, so, with the sample covariance matrix $S := \frac{1}{D} \sum_{d=1}^{D} (x_d - \bar{x})(x_d - \bar{x})^T$

$$J = \frac{1}{D} \sum_{d=1}^{D} \|x_d - \tilde{x}_d\|^2 = \frac{1}{D} \sum_{d=1}^{D} \sum_{j=K+1}^{V} ((x_d - \bar{x})^T u_j)^2 = \frac{1}{D} \sum_{j=K+1}^{V} \sum_{d=1}^{D} u_j^T (x_d - \bar{x})(x_d - \bar{x})^T u_j$$

$$= \sum_{j=K+1}^{V} u_j^T S u_j$$
Maybe we can get away with linear algebra?

Principal Component Analysis

Beltrami, 1873, Jordan, 1874, Pearson, 1901, Schmidt, 1907, Hotelling, 1933, Lanczos, 1950

To find a set of *orthonormal* vectors $u_i$ to minimize the square reconstruction error

$$ J = \frac{1}{D} \sum_{d=1}^{D} \|x_d - \tilde{x}_d\|^2 = \sum_{j=K+1}^{V} u_j^T S u_j $$

Choose $U$ as the eigenvectors of the sample covariance $S := \frac{1}{D} \sum_{d=1}^{D} (x_d - \bar{x})(x_d - \bar{x})^T$, and get the best rank $K$ reconstruction $\tilde{x}_d$ by setting

$$ \tilde{x}_d := \sum_{k=1}^{K} a_{dk} u_k + \sum_{j=K+1}^{V} b_j u_j = \sum_{i=1}^{M} (x_d^T u_i) u_i + \sum_{i=M+1}^{D} (\bar{x}^T u_i) u_i $$

This yields $J = \sum_{j=K+1}^{V} \lambda_j$ (where $\lambda_j$ are the eigenvalues of $S$, sorted descendingly). If we first center the data $\hat{X} = X - 1x^T$, so $b = 0$, the $U$ are the (right) singular vectors of $\hat{X} = Q\Sigma U^T$. 
Probabilistic PCA

a maximum-likelihood derivation


Treat the loss, up to scaling, as a non-normalised negative log likelihood:

\[ J = -c \cdot \log p(X | \tilde{X}) + \log Z = \frac{1}{D} \sum_{d=1}^{D} ||x_d - \tilde{x}_d||^2 \]

\[ \Rightarrow p(X | \tilde{X}) = \prod_{d=1}^{D} \mathcal{N}(x_d; \tilde{x}_d, \sigma^2 I) \]

We also need to encode that we want a low-dimensional, linear embedding, and that the embedding should be in terms of independent (orthogonal) dimensions.
Thus, consider

\[ x_d = Va_d + \mu + \epsilon \quad \text{with} \quad p(a_d) = \mathcal{N}(0; I_K), \quad V \in \mathbb{R}^{V \times K} \text{ and } p(\epsilon) = \mathcal{N}(0; \sigma^2) \]

with marginal likelihood (where \( C := VV^T + \sigma^2 I \))

\[
p(X) = \int \prod_{d=1}^{D} p(x_d | a_d) p(a_d) \, da_d = \prod_{d} \mathcal{N}(x_d; \mu, C)
\]

\[
\log p(X) = -\frac{DV}{2} \log(2\pi) - \frac{D}{2} \log |C| - \frac{1}{2} \sum_{d=1}^{D} (x_d - \mu)^\top C^{-1} (x_d - \mu)
\]

\[ \bar{x} = \arg \max_{\mu} \log p(X), \quad \text{thus the max. lik. can be written as} \]

\[
\log p(X) = -\frac{D}{2} \left( V \log(2\pi) + \log |C| + \text{tr}(C^{-1} S) \right)
\]
Probabilistic PCA

a maximum-likelihood derivation

\[ \log p(X) = -\frac{D}{2} (V \log(2\pi) - \log |C| + \text{tr}(C^{-1}S)) \]

yields max. lik. for \( V, \sigma^2 \) at [Tipping & Bishop, 1999], with \( RR^\top = I_K \) and \( S = U\Lambda U^\top \)

\[ V_{ML} = U_{1:K}(\Lambda_K - \sigma^2 I)^{1/2}R \quad \text{and} \quad \sigma^2_{ML} = \frac{1}{V - K} \sum_{j=K+1}^V \lambda_j \]

setting \( \sigma^2, \mu, U \) this way, and \( R = I \) w.l.o.g., gives posterior

\[
\begin{align*}
    p(a_d | x_d) &= \mathcal{N}(a_d; (V^\top V + \sigma^2 I)^{-1}V^\top(x_d - \bar{x}), \sigma^2(V^\top V + \sigma^2 I)^{-1}) \\
    &= \mathcal{N}(a_d; \Lambda_K^{-1}(\Lambda_K - \sigma^2 I_K)^{1/2}U_{1:K}(x_d - \bar{x}), \sigma^2 \Lambda^{-1})
\end{align*}
\]
So, does it work?

```python
count_vect_lsa = CountVectorizer(max_features=VOCAB_SIZE, stop_words=['000'])
X_count = count_vect_lsa.fit_transform(preprocessed).toarray()
U_, S_, V_T_ = np.linalg.svd(X_count, full_matrices=False)
```
Latent Semantic Indexing / Principal Component Analysis

a first result on our dataset

1. tonight fight taxis faith century today enemy fellow
2. year program world new work need help america
3. dollar war program fiscal year expenditure million united
4. man law dollar business national corporation legislation labor
5. administration policy energy program continue development provide effort
6. war nation power man mexico world peace public
7. united war states american world mexico man nation
8. government people states world free shall dollar constitution
9. year free nation world increase report subject great
10. world free gold government bank note american treasury

Probabilistic ML — P. Hennig, SS 2021 — Lecture 19: The Sum-Product Algorithm — © Philipp Hennig, 2021 CC BY-NC-SA 3.0
The singular value decomposition (SVD) minimizes \( \|X - Q\Sigma U^\prime\|_F^2 \) for orthonormal matrices \( Q \in \mathbb{R}^{D \times K} \) and \( U \in \mathbb{R}^{V \times K} \), and a diagonal \( \Sigma \in \mathbb{R}^{K \times K} \) with positive diagonal entries (the singular values).

We might naïvely think of \( Q \) as a mapping from documents to topics, \( U^\prime \) from topics to words, and \( \Sigma \) as the relative strength of topics.

However, there are several problems:

- the matrices \( Q, U \) returned by the SVD are in general dense: Every document contains contributions from every topic, and every topic involves all words.
- the entries in \( Q, U, \Sigma \) are hard to interpret: They do not correspond to probabilities
- the entries of \( Q, U \) can be negative! What does it mean to have a negative topic?
We need Sparsity

How about one topic per document?

For PCA, we allowed $Z \in \mathbb{R}^{D \times K}$. Maybe we need $Z \in \{0; 1\}^{D \times K}$ and $Z1_K = 1_D$?
Mixture Models

generative modelling with discrete classes

a supervised problem that can be solved discriminatively in a linear fashion
Mixture Models

generative modelling with discrete classes

a *supervised* problem that can be solved *discriminatively* in a *nonlinear* fashion

---

Probabilistic ML — P. Hennig, SS 2021 — Lecture 19: The Sum-Product Algorithm — © Philipp Hennig, 2021 CC BY-NC-SA 3.0
a **supervised** problem that can be solved **generatively** (in a Gaussian fashion?)
Mixture Models

generative modelling with discrete classes

an unsupervised problem

https://www.stat.cmu.edu/larry/all-of-statistics/data/faithful.dat

Mixture Models

generative modelling with discrete classes

a Gaussian mixture

\[
p(x_d, Z) = \prod_{d=1}^{D} p(z_d \mid \pi) p(x_d \mid z_d, \mu, \Sigma) = \prod_{d=1}^{D} \prod_{k=1}^{K} \pi^z_{dk} N(w_d; \mu_k, \Sigma_k)^{z_{dk}}
\]
A Gaussian Mixture isn’t quite right yet

word counts aren’t real-valued
A Mixture of Probabilities?

Desiderata

- topics should be probabilities: \( p(x_d \mid k) = \prod_{v=1}^{V} \theta_{kv}^{x_dv} \)
- but documents should have several topics! Let \( \pi_{dk} \) be the probability to draw a word from topic \( k \)