FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG

SCHOOL OF ENGINEERING

Lecture Pattern Analysis

Part 11: Inference via Sampling

Christian Riess
IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg

June 21, 2022

Introduction

- Recall that inference in a Bayesian framework with observed, unobserved, and hidden variables may easily be analytically intractable
- However, approximate solutions may be well tractable
- Sampling-based approximations are an alternative to variational inference

- We will specifically look at Gibbs sampling to fit a GMM and to simultaneously select the number of GMM components
- Gibbs sampling is a special case of the Metropolis-Hastings approach to Markov Chain Monte Carlo sampling[1]
- It oftentimes requires simpler sampling techniques as submodules, specifically the Adaptive Rejection Sampling (ASR)

[1] This lecture refers to Bishop Sec. 11–11.1.3, Sec. 11.2.1, Sec. 11.3, and the paper by Rasmussen, which can be found on studOn

Application Example: Evaluation of Expectations

- Sampling-based methods are particularly useful to approximate expectations
- If the expectation

$$\mathbb{E}[f] = \int f(\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z} \tag{1}$$

is analytically intractable, then draw L samples from $p(\mathbf{z})$ and calculate

$$\hat{f} = \frac{1}{L}\sum_{l=1}^{L} f(\mathbf{z}^{(l)}) \tag{2}$$

- This estimator is unbiased, in the sense that $\mathbb{E}[\hat{f}] = \mathbb{E}[f]$ and the variance is

$$\mathrm{var}[\hat{f}] = \frac{1}{L}\mathbb{E}\left[(f - \mathbb{E}[f])^2\right] \tag{3}$$

- Also note that the accuracy of the estimator does not depend on the dimensionality of **z**, i.e., few samples may suffice
- Other application examples are inference in graphical models (more later)

Standard Distributions and Criticism on the “Lecture-1-Sampler”

- Analytic mappings from uniform distributions to other distributions exist for
  - Gaussian distributions
  - Exponential distributions $p(y) = \lambda \exp(-\lambda y)$
  - Cauchy distributions $p(y) = \frac{1}{\pi}\frac{1}{1+y^2}$
- Hence, sample $p(z)$ from a uniform distribution, and transform these samples to the target distribution $p(y)$ through some function $y = f(z)$
- Note that you need to include the derivative (1-D) or the Jacobian ($>$ 1-D):

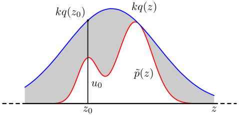
$$p(y) = p(z)\left|\frac{\mathrm{d}z}{\mathrm{d}y}\right| \qquad p(y_1, \ldots, y_M) = p(z_1, \ldots, z_M)\left|\frac{\partial(z_1, \ldots, z_M)}{\partial(y_1, \ldots, y_M)}\right| \tag{4}$$

- Our sampler from Lecture 1 is more general, but it requires to explicitly know the full PDF, which is quite restrictive
- For our inference task, we need a sampler for **non-standard distributions** that does **not require knowledge of the full PDF**

Rejection Sampling

- Assume that you can evaluate $p(z)$ up to normalization, i.e., $p(z) = \frac{1}{Z_p}\tilde{p}(z)$
- Assume that you have a simpler distribution $q(z)$ and a constant k, s.t. $kq(z) \geq \tilde{p}(z)$ for all z
- Then, sample in two steps:

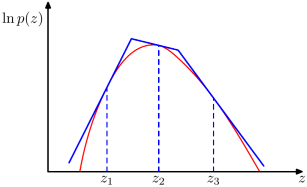
  1. Draw z_0 from $q(z)$
  2. Draw u_0 from a uniform distribution over$[0, kq(z_0)]$
  3. Reject (z_0, u_0) if $u_0 > \tilde{p}(z_0)$, otherwise keep z_0



- The method is correct, since
  - prior to rejection, the pair (z_0, u_0) is uniformly distributed across the area of the curve $kq(z)$
  - after rejection, the pair (z_0, u_0) is uniformly distributed across the area of the curve $\tilde{p}(z)$

Adaptive Rejection Sampling

- Rejection sampling becomes inefficient if q and p differ too much
- However, a better-fitting envelope q might not have a simple analytic form
- Adaptive Rejection Sampling (ASR) constructs q ad-hoc from $p(z)$
- This works particularly well on log-concave functions, i.e., where derivatives of $\log p(z)$ are non-increasing functions of z



- Fitting a set of lines to the log of the function is equivalent to fitting a piecewise exponential distribution to the original function, i.e.,

$$q(z) = k_i \lambda_i \exp\{-\lambda_i (z - z_i)\} \qquad \hat{z}_{i-1,i} < z \leq \hat{z}_{i,i+1} \tag{5}$$

Markov Chain Monte Carlo Sampling

- In high-dimensional spaces, the gap between q and $\tilde{p}$ increases
- Rejection sampling and ASR do not perform well then, due to excessive rejections

- Markov Chain Monte Carlo (MCMC) is much better in high-dimensional spaces
- The idea is to sample from some kind of state space $\mathbf{z}^{(\tau)}$ that takes the previous samples $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(\tau-1)}$ into account (resulting in a random walk)
- A Markov Chain, specifically, models only first-order statistical dependencies

$$p\left(\mathbf{z}^{(\tau+1)} | \mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(\tau)}\right) = p\left(\mathbf{z}^{(\tau+1)} | \mathbf{z}^{(\tau)}\right) \tag{6}$$

- A famous MCMC algorithm is the Metropolis-Hastings method, but we jump right to an important special case, namely Gibbs Sampling

Gibbs Sampling

- We aim to sample from the distribution $p(\mathbf{z}) = p(z_1, \ldots z_M)$ of M random variables (which are somehow initialized)
- Each step of Gibbs sampling updates one variable by drawing from the distribution of that variable conditioned on the others, i.e.,

```
1. Initialize $\{z_i : i = 1, \ldots, M\}$
2. For $\tau = 1, \ldots, T$:
    2.1 Sample $z_1^{(\tau+1)} \sim p\left(z_1 | z_2^{(\tau)}, \ldots, z_M^{(\tau)}\right)$
    2.2 Sample $z_2^{(\tau+1)} \sim p\left(z_2 | z_1^{(\tau)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)}\right)$
        ⋮
    2.M Sample $z_M^{(\tau+1)} \sim p\left(z_M | z_1^{(\tau)}, \ldots, z_{M-1}^{(\tau)}\right)$
```

- Note that subsequent samples are correlated (due to the Markov chain)
- However, we will often seek only one single reasonable representative, e.g., one single fitted GMM model

Gibbs Sampling applied to GMM Models

- Let us quickly browse through Rasmussen's paper on infinite GMMs, i.e., GMMs with automatic selection of the number of components
- Without going into details, you will notice that posteriors are created for
  - all GMM parameters (μ_i, s_i, π_i),
  - the indicator variables c_i, and
  - the priors for creating new GMM components λ, r, w, β, α
- Effectively, one sample from this Gibbs sampler is a full GMM.
  Fig. 2 (right) shows how these sampled GMMs vary in size between 15 and 25 components
- We will see another application when doing inference for Markov Random Fields

FRIEDRICH-ALEXANDER-
UNIVERSITÄT
ERLANGEN-NÜRNBERG

SCHOOL OF ENGINEERING

Lecture Pattern Analysis

Part 12: Curse of Dimensionality

Christian Riess

IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg

June 21, 2022

Introduction

- So far, we looked at low-dimensional feature vectors (also to visualize results)
- However, real data oftentimes consists of 100s of dimensions
- Generally speaking, the difficulty of all data analysis tasks increases with data dimensionality
- This increase in difficulty is sometimes referred to as “Curse of Dimensionality” (Bellman, 1961)
- In this lecture, we illustrate three difficulties associated with high-dimensional data[1]
- This motivates the dimensionality reduction / manifold learning in the next lectures

[1] The content of this lecture refers to Bishop Sec. 1.4

Difficulty 1: Visualization

- For the understanding of the data, it is most useful if it can be visualized
- However, data is more than often very high-dimensional
- Examples:
  - Remote sensing is the research field of processing satellite recordings, e.g., for environmental or agricultural monitoring.
    Photographs of the earth surface are not done in RGB, but in hundreds much more narrow color bands
  - Deep neural networks learn feature maps (“representations”) with dozens to hundreds of dimensions
    How can we plausibly demonstrate that the learned representation maps similar objects to similar locations in the feature space?
  - The success of Netflix, amazon, google, etc. critically depends on making the most tempting next recommendation to customers
    How to look into improvements of such a recommendation system, given millions of mutually different individual consumption histories?

Difficulty 2: Statistical Space Subdivision

- Consider the fundamental assumption of pattern recognition that similar features are at similar locations in the sample space
- Hence, a classifier or regressor must make local predictions
- However, assume (for simplicity) equally-sized cells:
  their number grows exponentially with the dimensions

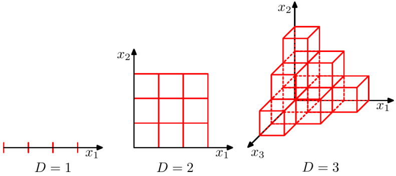


Figure 1.21 Illustration of the curse of dimensionality, showing how the number of regions of a regular grid grows exponentially with the dimensionality D of the space. For clarity, only a subset of the cubical regions are shown for $D = 3$.

- Hence, we require more model parameters, and moreover an exponentially growing number of data points for sufficient observations per cell
  (our kernel estimators will have particular difficulties in high dimensions)

Difficulty 3: Distances become Less Discriminative

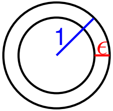
- Consider a D-dim. sphere with radius 1 with uniformly distributed samples
- The volume of that sphere in dependency of the radius r is

$$V_D(r) = K_D \cdot r^D \tag{1}$$

  where K_D is a constant volume factor
- The fraction $f_D(\epsilon)$ of data at the boundary between $V_D(1)$ and $V_D(1-\epsilon)$ is:

$$f_D(\epsilon) = \frac{V_D(1) - V_D(1-\epsilon)}{V_D(1)} = \frac{1-(1-\epsilon)^D}{1} = 1-(1-\epsilon)^D \tag{2}$$

- Interestingly, $f_D(\epsilon)$ rapidly approaches 1, e.g.,
  $D = 10$, $\epsilon = 0.1$: $f_{10}(0.1) = 65\%$
  $D = 100$, $\epsilon = 0.01$: $f_{100}(0.01) = 63\%$



- When most samples lie at the boundary, the distances between samples become more similar, and hence the distances become less meaningful
- This issue also affects rejection sampling in high dimemsions

Lecture Pattern Analysis

Part 13: Principal Component Analysis

Christian Riess

IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg

June 21, 2022

Introduction

- Principal Component Analysis (PCA), a.k.a. “Karhunen-Loeve Transform” or “KL-Transform” is a workhorse all across science and engineering
- PCA provides a more compact representation in a lower-dim. space
- Brief overview[1]:
  - PCA is a linear projection onto an orthogonal basis **U**, i.e.,

$$\mathbf{u}_i^\mathrm{T} \cdot \mathbf{u}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

  - This basis are the eigenvectors of the (mean-free) data covariance
  - Thus, the calculation of PCA is essentially to normalize the data and to perform an eigenvalue decomposition
  - The magnitude of the eigenvalues indicates the contribution of a dimension to the covariance of the data

[1] The literature source for this lecture is Bishop Sec. 12.1.1

Objective Function and Normalization

- Core idea: find a linear mapping $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, $d' \ll d$, that maximizes the variance (spread) of the data along each dimension
- Objective function:

$$J = \sum_{i,j=1}^{N} (\Phi \mathbf{x}_i - \Phi \mathbf{x}_j)^\mathsf{T} (\Phi \mathbf{x}_i - \Phi \mathbf{x}_j) + \lambda(\Phi^\mathsf{T} \Phi - 1) \tag{2}$$

  where $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$ are the data points
- Assume zero-mean samples. Hence, in practice, subtract the mean of the samples to obtain

$$\sum_{i=1}^{N} \mathbf{x}_i = 0 \tag{3}$$

Derivation of the Principal Components

- We seek a projection **u** onto the 1-D subspace that maximizes the variance, and show that **u** is the largest eigenvector of the covariance matrix
- **u** is the first column of Φ, further vectors are obtained by induction:
  - Project the data on the $d - 1$-dim. subspace orthogonal to **u**
  - Repeat the reasoning $d' - 1$ times
- To begin, let $\mathbf{u} \in \mathbb{R}^d$ be an arbitrary direction of unit length, i.e., $\mathbf{u}^\mathsf{T}\mathbf{u} = 1$
- The inner product $\mathbf{u}^\mathsf{T}\mathbf{x}$ projects **x** onto a 1-D space
- The variance of the projected data is

$$\frac{1}{N}\sum_{i=1}^{N}(\mathbf{u}^\mathsf{T}\mathbf{x}_i - \mathbf{u}\bar{\mathbf{x}})^2 = \mathbf{u}^\mathsf{T}\mathbf{S}\mathbf{u} \tag{4}$$

where $\bar{\mathbf{x}}$ is the (component-wise) mean of all $\mathbf{x}_i$ and **S** is the covariance matrix

$$\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\mathsf{T} \tag{5}$$

Maximizing the Variance

- We seek a unit-length direction **u** that maximizes the variance:

$$\mathbf{u}^{\mathsf{T}}\mathbf{S}\mathbf{u} + \lambda(1 - \mathbf{u}^{\mathsf{T}}\mathbf{u}) \rightarrow \max \tag{6}$$

where λ is a Lagrange multiplier to include the constraint $\mathbf{u}^{\mathsf{T}}\mathbf{u} = 1$

- The maximum is found by calculating the derivative w.r.t. **u**, and to set the equation equal to 0:

$$\frac{\partial}{\partial \mathbf{u}} \mathbf{u}^{\mathsf{T}}\mathbf{S}\mathbf{u} + \lambda(1 - \mathbf{u}^{\mathsf{T}}\mathbf{u}) \overset{!}{=} 0 \tag{7}$$

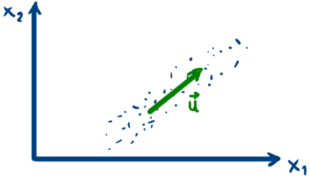
$$\Leftrightarrow \quad 2\mathbf{S}\mathbf{u} = 2\lambda\mathbf{u} \tag{8}$$

$$\Leftrightarrow \quad \mathbf{S}\mathbf{u} = \lambda\mathbf{u} \tag{9}$$

- This is just the eigenvector decomposition of **S**. Hence, the eigenvector associated with the largest eigenvalue provides maximum covariance
- This vector is called a “principal component”

Remarks

- You probably know sketches of the direction of maximum covariance:



- The relative magnitude of an eigenvalue indicates the percentage of variance that is represented. For example, if

$$\left(\sum_{i=1}^{d'} \lambda_i\right) \Big/ \left(\sum_{i=1}^{d} \lambda_i\right) = 0.98 \ , \qquad (10)$$

then a d'-dim. subspace preserves 98$\%$ of the variance of the data

- This argument is used, e.g., in remote sensing to compress 100s of (correlated) color bands to less than 10 dimensions