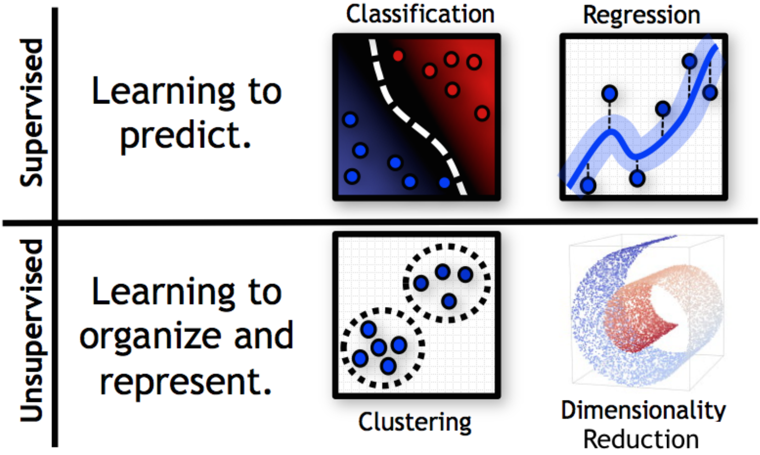
COMPSCI 689
Lecture 15: Joint Probability Models

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Machine Learning Tasks



Probabilistic Unsupervised Learning

Basic Definitions:

- Input: $\mathbf{X} = [X_1, ..., X_D] \in \mathcal{X} = \mathcal{X}_1 \times ... \times \mathcal{X}_D$
- True Distribution: $P_*(\mathbf{X} = \mathbf{x}) = P_*(\mathbf{x})$
- Parametric Model: $P(\mathbf{X} = \mathbf{x}|\theta) = P(\mathbf{x}|\theta)$

In probabilistic unsupervised learning, our goal is to find a model $P(\mathbf{x}|\theta)$ that is as close as possible to $P_*(\mathbf{x})$.

Losses for Distributions

Unlike in supervised learning, there are few commonly used losses between distributions:

- Absolute Loss: $L_1(P_* \| P_\theta) = \mathbb{E}_{P_*(\mathbf{X})}\left[|P_*(\mathbf{x}) - P(\mathbf{x}|\theta)|\right]$
- Squared Loss: $L_2(P_* \| P_\theta) = \mathbb{E}_{P_*(\mathbf{X})}\left[(P_*(\mathbf{x}) - P(\mathbf{x}|\theta))^2\right]$
- KL Divergence: $KL(P_* \| P_\theta) = \mathbb{E}_{P_*(\mathbf{X})}\left[\log\left(\frac{P_*(\mathbf{x})}{P(\mathbf{x}|\theta)}\right)\right]$

Question: Which of these losses can we minimize using a sample of data $\mathcal{D} = \{\mathbf{x}_n\}_{1:N}$?

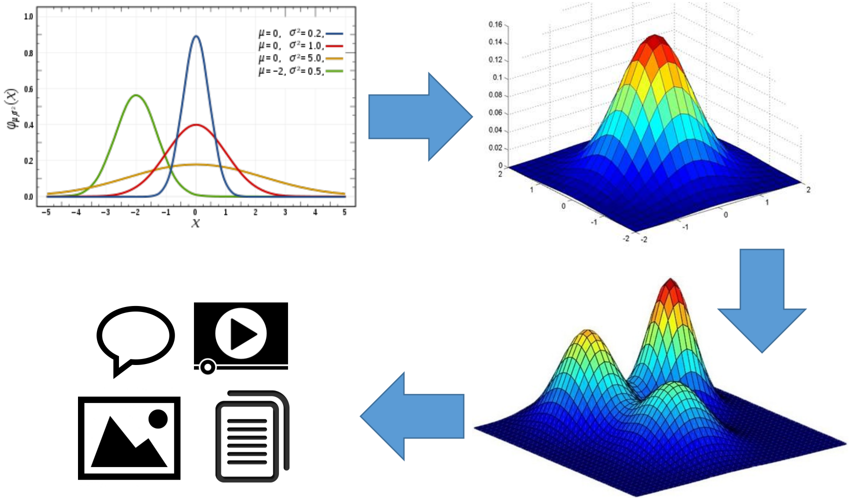
Optimizing KL Divergence

$$
\begin{aligned}
\min_\theta KL(P_* \| P_\theta) &= \min_\theta \int_{\mathcal{X}} P_*(\mathbf{x}) \big(\log P_*(\mathbf{x}) - \log P(\mathbf{x}|\theta) \big) d\mathbf{x} \\
&= \min_\theta \int_{\mathcal{X}} P_*(\mathbf{x}) \log P_*(\mathbf{x}) d\mathbf{x} - \int_{\mathcal{X}} P_*(\mathbf{x}) \log P(\mathbf{x}|\theta) d\mathbf{x} \\
&= \max_\theta \int_{\mathcal{X}} P_*(\mathbf{x}) \log P(\mathbf{x}|\theta) d\mathbf{x} \\
&\approx \max_\theta \int_{\mathcal{X}} P_{\mathcal{D}}(\mathbf{x}) \log P(\mathbf{x}|\theta) d\mathbf{x} \\
&= \max_\theta \frac{1}{N} \sum_{n=1}^{N} \log P(\mathbf{x}_n|\theta)
\end{aligned}
$$

Optimization-Based Unsupervised Learning

- As we can see, selecting the value of θ that makes the data the most likely is a Monte Carlo approximation to selecting the value of θ that minimizes $KL(P_*\|P_\theta)$.
- The dominant approaches to optimization-based unsupervised learning of probabilistic models are thus maximum likelihood estimation and its penalized/regularized derivatives, which are again equivalent to MAP estimation.
- Unsupervised learning with single random variables that follow standard distributions (Bernoulli, multinomial, Poisson, normal, exponential etc.) is easy using off-the-shelf MLE results.
- The interesting question is how to efficiently model complex distributions of many random variables?

Optimization-Based Unsupervised Learning

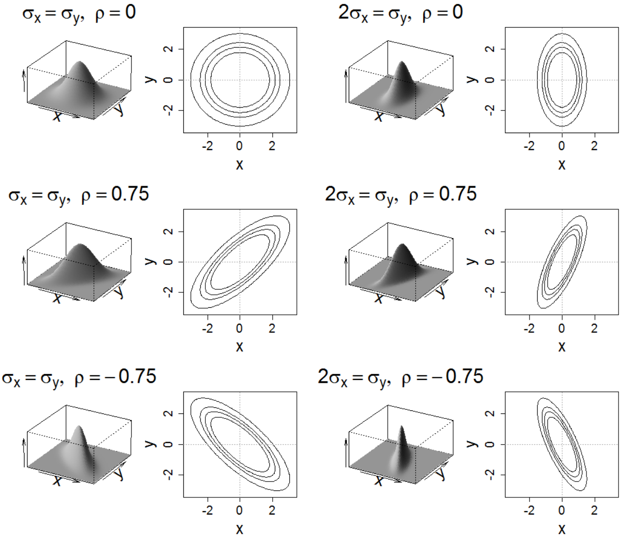


The Multivariate Normal

- The multivariate normal (or Gaussian) distribution is a fundamental building block for unsupervised learning with multiple real-valued random variables $\mathbf{X} \in \mathbb{R}^D$.
- The distribution has two parameters $\theta = [\mu, \Sigma]$. μ is the mean vector and Σ is the covariance matrix.
- We have $\mu \in \mathbb{R}^D$ and $\Sigma \in \mathbb{S}^D_+$, the space of symmetric, positive definite $D \times D$ matrices.
- The probability density is given below (assuming $\mathbf{x}$ and μ are column vectors):

$$\mathcal{N}(\mathbf{x}; \mu, \Sigma) = \frac{1}{|2\pi\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\Big)$$

Example: Bivariate Normal



MLE for the Multivariate Normal

- Given a data set $\mathcal{D} = \{\mathbf{x}_n\}_{1:N}$, the MLE for the multivariate normal is found by solving the optimization problem:

$$\mu^*, \Sigma^* = \arg\max_{\mu,\Sigma} \sum_{n=1}^{N} \log \mathcal{N}(\mathbf{x}_n; \mu, \Sigma)$$

- The solutions are:

$$\mu^* = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n, \qquad \Sigma^* = \frac{1}{N}\sum_{n=1}^{N} (\mathbf{x}_n - \mu^*)(\mathbf{x}_n - \mu^*)^T$$

Marginalization

- Suppose we have a joint distribution on a vector-valued random variable $\mathbf{X} \in \mathbb{R}^D$. Let $A \subseteq \{1, ..., D\}$, $M = |A|$, and $\mathbf{X}_A = [X_{A_1}, ..., X_{A_M}]$.

- The probability distribution $P(\mathbf{X}_A = \mathbf{x}_A)$ is called the *marginal distribution* of $\mathbf{X}_A$.

- Let $B = \{1, ..., D\}/A$. The marginal distribution of $\mathbf{X}_A$ is then given by:

$$P(\mathbf{X}_A = \mathbf{x}_A) = \int_{\mathcal{X}_B} P(\mathbf{X}_A = \mathbf{x}_A, \mathbf{X}_B = \mathbf{x}_B) d\mathbf{x}_B$$

Marginalization for MVNs

- The multivariate normal distribution has the remarkable (and convenient) property of being closed under marginalization.

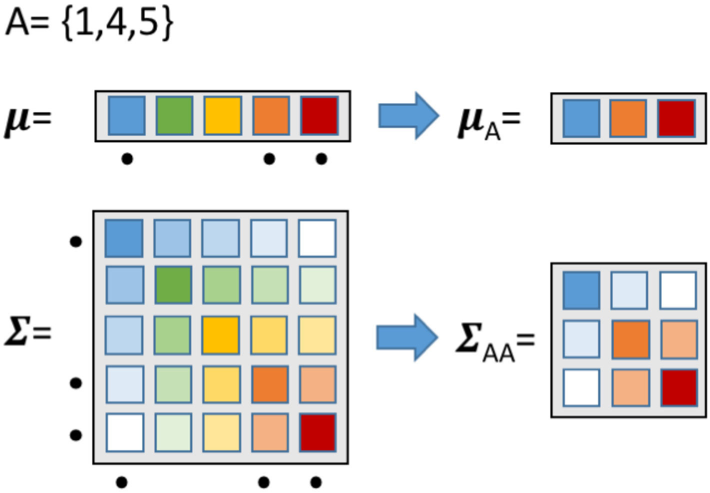
- Suppose we have an MVN $P(\mathbf{X}|\theta) = \mathcal{N}(\mathbf{X}; \mu, \Sigma)$ for $\mathbf{X} \in \mathbb{R}^D$. Let $A \subseteq \{1, ..., D\}$, $B = \{1, ..., D\}/A$, and $M = |A|$. We have:

$$P(\mathbf{X}_A = \mathbf{x}_A) = \mathcal{N}(\mu_A, \Sigma_{AA})$$

  where $\mu_A = [\mu_{A_1}, ..., \mu_{A_M}]$ and $(\Sigma_{AA})_{ij} = \Sigma_{A_i, A_j}$.

- In other words, we get the marginal distribution on a subset of $\mathbf{X}$ just by discarding the elements of μ that correspond to B, and the rows and columns of Σ that correspond to B.

Marginalization for MVNs: Example



Conditioning

- Suppose we have a joint distribution on a vector-valued random variable $\mathbf{X} \in \mathbb{R}^D$. Let $A \subseteq \{1, ..., D\}$ and let $B = \{1, ..., D\}/A$.

- The *conditional distribution* of $\mathbf{X}_A$ given $\mathbf{X}_B$ is defined as shown below:

$$P(\mathbf{X}_A = \mathbf{x}_A | \mathbf{X}_B = \mathbf{x}_B) = \frac{P(\mathbf{X}_A = \mathbf{x}_A, \mathbf{X}_B = \mathbf{x}_B)}{P(\mathbf{X}_B = \mathbf{x}_B)}$$

- This definition follows from the definition of conditional probability for events.

- Note that the numerator is the joint distribution and the denominator is the marginal distribution of $\mathbf{X}_B$.

Conditioning for MVNs

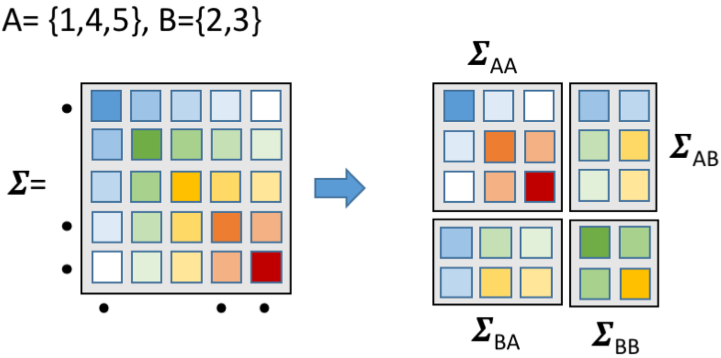
- The multivariate normal distribution has the remarkable (and convenient) property of also being closed under conditioning.
- Suppose we have an MVN $P(\mathbf{X}|\theta) = \mathcal{N}(\mathbf{X}; \mu, \Sigma)$ for $\mathbf{X} \in \mathbb{R}^D$. Let $A \subseteq \{1, ..., D\}$, $B = \{1, ..., D\}/A$. We have:

$$P(\mathbf{X}_A = \mathbf{x}_A | \mathbf{X}_B = \mathbf{x}_B) = \mathcal{N}(\mathbf{x}_A; \mu_{A|B}, \Sigma_{AA|B})$$

$$\mu_{A|B} = \mu_A + \Sigma_{AB}(\Sigma_{BB})^{-1}(\mathbf{x}_B - \mu_B)$$

$$\Sigma_{AA|B} = \Sigma_{AA} - \Sigma_{AB}(\Sigma_{BB})^{-1}\Sigma_{BA}$$

Conditioning for MVNs: Example



Posterior Predictions

- The significance of marginalization and conditioning in multivariate joint distributions is that they allow us to observe any subset of the variables B, and make predictions about any other subset A.
- In particular, conditioning in an MVN can be used to provide a regression output $\hat{x}_A$ for any single random variable in $\mathbf{X}$ using:

$$\hat{x}_A = \mu_A + \Sigma_{AB}(\Sigma_{BB})^{-1}(\mathbf{x}_B - \mu_B)$$

The MVN model can be thought of as encoding an exponential number of different linear regression models with a quadratic number of parameters.

The Problem With General Joint Distributions

- The multivariate normal distribution is only applicable to real-valued data and makes a number of very strong assumptions.
- Most other basic continuous random variables lack tractable extensions to joint distributions over many variables.
- A finite collection of finite discrete random variables always has a joint distribution that can be represented as a look-up table with one row for each joint configuration in $\mathcal{X}$.
- However, if $\mathbf{X} = [X_1, ..., X_D]$, then $|\mathcal{X}| \geq 2^D$. This makes directly learning discrete joint distributions intractable for even moderate D.

Example: Finite Joint Discrete Joint Distributions

Consider the case where $\mathbf{X} \in \{0, 1\}^5$. How large is $P(\mathbf{X})$?

x	P(X=x|θ)
00000	θ_0
00001	θ_1
00010	θ_2
00011	θ_3
$\vdots$	
11111	θ_{31}

Structured Probability Models

- One solution to these problems is to use structured probability distributions that can be learned efficiently and have many fewer parameters.

- The primary mathematical tools are the chain rule of probability and probabilistic independence.

Chain Rule

- The Chaine Rule of Probability states that:

$$P(X_1, ..., X_D) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\cdots P(X_D|X_1, ..., X_{D-1})$$

- This result holds for any permutation of the indices $1, ..., D$.
- It is derived from repeated application of the product rule $P(\mathbf{X}_A, \mathbf{X}_B) = P(\mathbf{X}_A|\mathbf{X}_B)P(\mathbf{X}_B)$, which is in turn derived from the conditional probability rule.

Marginal Independence

$$\mathbf{X} \perp \mathbf{Y} \iff P(\mathbf{X}|\mathbf{Y}) = P(\mathbf{X})$$

$$\mathbf{X} \perp \mathbf{Y} \iff P(\mathbf{Y}|\mathbf{X}) = P(\mathbf{Y})$$

$$\mathbf{X} \perp \mathbf{Y} \iff P(\mathbf{Y}, \mathbf{X}) = P(\mathbf{X})P(\mathbf{Y})$$

Conditional Independence

$$\mathbf{X} \perp \mathbf{Y} | \mathbf{Z} \iff P(\mathbf{X}|\mathbf{Y}, \mathbf{Z}) = P(\mathbf{X}|\mathbf{Z})$$

$$\mathbf{X} \perp \mathbf{Y} | \mathbf{Z} \iff P(\mathbf{Y}|\mathbf{X}, \mathbf{Z}) = P(\mathbf{Y}|\mathbf{Z})$$

$$\mathbf{X} \perp \mathbf{Y} | \mathbf{Z} \iff P(\mathbf{Y}, \mathbf{X}|\mathbf{Z}) = P(\mathbf{X}|\mathbf{Z})P(\mathbf{Y}|\mathbf{Z})$$

Compactness from Independence

Suppose we have a joint distribution $P(A, B, C)$ and we know that the independence relation $A \perp B | C$ holds. How can we exploit this fact to simplify $P(A, B, C)$?

- Chain Rule: $P(A, B, C) = P(A|B, C)P(B|C)P(C)$
- Conditional Independence: $A \perp B | C \rightarrow P(A|B, C) = P(A|C)$
- Simplification: $P(A, B, C) = P(A|C)P(B|C)P(C)$

Structured probability models such as *Bayesian network* use exactly this approach to simplify a joint distribution. We will look as special cases of this general model class.