COMPSCI 689
Lecture 17: (Generalized) Latent Linear Models

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

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

Mixture Models

- A mixture model is a probabilistic clustering model with the following marginal distribution over the observed data:

$$P(\mathbf{X} = \mathbf{x}|\theta) = \sum_{z=1}^{K} P(\mathbf{X} = \mathbf{x}|Z = z, \theta_z)P(Z = z|\pi)$$

- Here, $Z \in \{1, ..., K\}$ is a discrete random variables that takes one of K discrete values.

Mixture Models and Clusters

- Mixture models are well suited to modeling data that matches the cluster assumption.
- The parameters of basic mixture component distributions $P(\mathbf{X} = \mathbf{x}|Z = z, \theta_z)$ can typically be thought of as “prototypes" for data generated from the corresponding cluster.
- In a Gaussian mixture model, for example, data are generated by selecting a mixture component z, and then adding (correlated) noise to the mean parameters.
- This generates clumps of data around the mean of each component.

Mixture Models and Manifolds

- When data instead fall on a low dimensional manifold instead of around a discrete collection of prototypes, mixture models can still be used as universal density models.
- However, it may take a very large number of mixture components to adequately approximate a probability density defined on a manifold.
- In these cases, models designed specifically for manifolds typically give better results.
- We will begin with the case of linear manifolds.

Factor Analysis

- Factor analysis is a classical statistical model for linear manifolds based on the multivariate normal distribution.
- The model asserts that real-valued data $\mathbf{x} \in \mathbb{R}^D$ are generated in a two stage process that starts by first generating a low-dimensional latent factor vector $\mathbf{z} \in \mathbb{R}^K$ from a multivariate normal distribution.
- The observed $\mathbf{x}$'s are then generated by a linear combination of basis vectors weighted by the latent factor values: $\mathbf{Wz}$ with independent Gaussian noise added.

Factor Analysis: Probabilistic Model

- The probabilistic model for factor analysis is shown below:

$$P(\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z}) = P(\mathbf{X} = \mathbf{x} | \mathbf{Z} = \mathbf{z}) P(\mathbf{Z} = \mathbf{z})$$
$$P(\mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{z}; \mu_0, \Sigma_0)$$
$$P(\mathbf{X} = \mathbf{x} | \mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z} + \mu, \Psi)$$

- We typically assume that the latent mean $\mu_0 = 0$, and that the latent covariance matrix Σ_0 is the identity matrix.
- We also typically assume that Ψ is a postive, diagonal matrix and that the data set mean has been removed so that $\mu = 0$.

Factor Analysis: Marginal Distribution

- The marginal distribution of **X** is is given by:

$$\begin{aligned} P(\mathbf{X} = \mathbf{x}) &= \int \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z}, \Psi)\mathcal{N}(\mathbf{z}; 0, I)d\mathbf{z} \\ &= \mathcal{N}(\mathbf{x}; 0, \mathbf{W}\mathbf{W}^T + \Psi) \end{aligned}$$

- Thus, to learn the factor analysis model, we need to maximize the log marginal likelihood:

$$\mathcal{L}(\mathcal{D}, \theta) = -\frac{N}{2}\log(|2\pi(\mathbf{W}\mathbf{W}^T + \Psi)|) - \frac{1}{2}\mathbf{x}_n^T(\mathbf{W}\mathbf{W}^T + \Psi)^{-1}\mathbf{x}_n$$

Factor Analysis: Learning

- As with mixture models, we can simply maximize the sum of the log marginal likelihoods over a data set to learn the model parameters μ, Ψ and $\mathbf{W}$. However, we can again obtain an EM algorithm based on Jensen's inequality and lower bound optimization.

- The E-step again requires the posterior over the latent variables $q_n(\mathbf{z}) = P(\mathbf{Z} = \mathbf{z}|\mathbf{X} = \mathbf{x}_n)$. We have:

$$
\begin{aligned}
P(\mathbf{z}|\mathbf{x}_n) &= \mathcal{N}(\mathbf{z}; \mathbf{m}_n, \Sigma_n) \\
\Sigma_n &= (I + \mathbf{W}^T \Psi \mathbf{W})^{-1} \\
\mathbf{m}_n &= \Sigma_n \mathbf{W}^T \Psi^{-1} (\mathbf{x}_n - \mu)
\end{aligned}
$$

Factor Analysis: Learning

- The M-Step updates are given by:

$$\mathbf{W} = \left(\sum_{n=1}^{N} \mathbf{x}_n \mathbf{m}_n^T\right) \left(\sum_{n=1}^{N} \mathbb{E}_{q_n}[\mathbf{z}\mathbf{z}^T | \mathbf{x}_n])\right)^{-1}$$

$$\Psi = \frac{1}{N} \text{diag} \left(\sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{W}\mathbf{m}_n)\mathbf{x}_n^T\right)$$

Latent Linear Models for Real Data

- The basic model architecture for factor analysis is shown below.

$$P(\mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{z}; \mu_0, \Sigma_0)$$
$$P(\mathbf{X} = \mathbf{x} | \mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z} + \mu, \Psi)$$

- The next question is, how can we model data that are not real-valued?
- One answer is to merge ideas from latent linear models with ideas from generalized linear models.

Generalized Latent Linear Models

- This model class retains the normal distribution on $\mathbf{Z}$, but models $\mathbf{X} = [X_1, ..., X_D]$ using generalized linear models.
- For example, we can model Binary data using a Bernoulli-Logistic model for $P(X_d = x_d|\mathbf{Z} = \mathbf{z})$:

$$\theta_{d|z} = \frac{1}{1 + \exp(-\mathbf{W}_d\mathbf{z})}$$
$$P(x_d|\mathbf{z}) = \theta_{d|z}^{x_d}(1 - \theta_{d|z})^{(1-x_d)}$$
$$P(\mathbf{x}|\mathbf{z}) = \prod_{d=1}^{D} P(x_d|\mathbf{z})$$

Exponential Family Factor Analysis

- In general, we can let $P(X_d = x_d | \mathbf{Z} = \mathbf{z})$ be an exponential family GLM of a type that matches x_d.
- The resulting model is called *Exponential Family Factor Analysis* or *Exponential Family Principal Components Analysis*.
- However, this model class has several significant issues that derive from the fact that $P(\mathbf{X} = \mathbf{x})$ can often not be computed analytically due to the intractability of the following integral:

$$P(\mathbf{X} = \mathbf{x}) = \int \prod_{d=1}^{D} P(\mathbf{x}_d | \mathbf{z}) \mathcal{N}(\mathbf{z}; 0, I) d\mathbf{z}$$

- As a result, $P(\mathbf{Z} | \mathbf{X} = \mathbf{x})$ is not computable and direct marginal likelihood maximization and exact EM break.

Learning Exponential Family Factor Analysis

- As a result, it is generally only possible to approximately learn an exponential family factor analysis model using maximum likelihood.

- Another approach is to modify the learning criteria. Instead of aiming to optimize $\mathcal{L}(\mathcal{D}, \theta) = \sum_{n=1}^{N} \log P(\mathbf{X} = \mathbf{x}_n | \theta)$, we can optimize the function below over $\mathbf{z}_{1:N}$ and θ:

$$\mathcal{J}(\mathcal{D}, \mathbf{z}_{1:N}, \theta) = \sum_{n=1}^{N} \log P(\mathbf{X} = \mathbf{x}_n, \mathbf{Z} = \mathbf{z}_n | \theta)$$

Learning Exponential Family Factor Analysis

- The loss function $\mathcal{J}(\mathcal{D}, \mathbf{z}_{1:N}, \theta)$ treats the latent variables $\mathbf{z}_{1:N}$ as another set of model parameters to learn and optimizes them.
- This removes the issues with integrating over $\mathbf{z}$, generally making the objective function easy to learn with off-the-shelf optimization tools.
- However, optimizing the $\mathbf{z}$'s instead of integrating over them has the downside of collapsing uncertainty.
- Effectively, learning concentrates on the most likely value of each $\mathbf{z}_n$ only.

Latent Linear Models

- The basic model architecture for generalized factor analysis is shown below.

$$P(\mathbf{Z}=\mathbf{z}) = \mathcal{N}(\mathbf{z};\mu_0,\Sigma_0)$$
$$P(\mathbf{X}=\mathbf{x}|\mathbf{Z}=\mathbf{z}) = \prod_{d=1}^{D} P(x_d|\mathbf{z})$$

- We can easily change the model to use different types of latent variables by changing $P(\mathbf{Z}=\mathbf{z})$ to a different probability distribution.

Latent Linear Models with Alternate Latent Variables

- As in factor analysis, we keep the latent variables independent of each other resulting in the following generalized model:

$$P(\mathbf{Z} = \mathbf{z}) = \prod_{k=1}^{K} P(z_k)$$

$$P(\mathbf{X} = \mathbf{x} | \mathbf{Z} = \mathbf{z}) = \prod_{d=1}^{D} P(x_d | \mathbf{z})$$

- For example, we can let $P(z_k) = \pi_k^{z_k}(1 - \pi_k)^{(1-z_k)}$ to obtain a model with binary latent factors.
- If we let $P(z_k)$ be a Laplace distribution and $P(x_d|\mathbf{z})$ be a normal distribution, the model is equivalent to Sparse Coding.

Latent Linear Models with Alternate Latent Variables

- For real-valued latent factors, the marginal likelihood and posterior may or may not be computable in closed form.
- When the marginal likelihood and posterior are not computable in closed form, optimization of the latent variables can be used (this is what the sparse coding learning algorithm does).
- A different problem arises with finite discrete latent variables (binary, categorical) where the marginal probability switches from an integral to a sum:

$$P(\mathbf{X} = \mathbf{x}) = \sum_{z_1 \in \mathcal{Z}} \cdots \sum_{z_K \in \mathcal{Z}} \prod_{d=1}^{D} P(x_d|\mathbf{z}) \prod_{k=1}^{K} P(z_k)$$

Latent Linear Models with Alternate Latent Variables

- This sum is computable in theory, but can take exponential time to compute.
- Note that we can't use the trick of optimizing the $\mathbf{z}$'s because that is now a discrete optimization problem that also requires exponential time.
- In this model class, with discrete $\mathbf{z}$'s, the posterior $P(\mathbf{Z}|\mathbf{X}=\mathbf{x})$ also typically takes exponential time to compute.
- For all of these reasons, learning in this model class is quite difficult and requires some significant approximations.
- As a result, other model structures are typically used when discrete latent variables are of interest.