Unifying Review of Linear Gaussian Models

Sam Roweis[1] and Zoubin Ghahramani[2]
Summarized by: Alejandro Santorum Varela

[1]Computation and Neural Systems, California Institute of Technology
Pasadena, CA, 91125, U.S.A.

[2]Department of Computer Science, University of Toronto
Toronto, Canada

January 11, 2021

Contents

General Model

Several unsupervised methods can be represented as instances of an underlying basic model, such us: factor analysis, SPCA, PCA, ...

General Model

Several unsupervised methods can be represented as instances of an underlying basic model, such us: factor analysis, SPCA, PCA, ...

Basic generative model

$$\mathbf{x_{t+1}} = \mathbf{A}\mathbf{x_t} + \mathbf{w_t} = \mathbf{A}\mathbf{x_t} + \mathbf{w_\bullet}, \qquad \mathbf{w_\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{1a}$$

$$\mathbf{y_t} = \mathbf{C}\mathbf{x_t} + \mathbf{v_t} = \mathbf{C}\mathbf{x_t} + \mathbf{v_\bullet}, \qquad \mathbf{v_\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{1b}$$

where $\mathbf{A} \in \mathcal{M}_{k\times k}$ (transition matrix) and $\mathbf{C} \in \mathcal{M}_{p\times k}$ (generative matrix).

General Model

Several unsupervised methods can be represented as instances of an underlying basic model, such us: factor analysis, SPCA, PCA, ...

Basic generative model

$$\mathbf{x_{t+1}} = \mathbf{A}\mathbf{x_t} + \mathbf{w_t} = \mathbf{A}\mathbf{x_t} + \mathbf{w_\bullet}, \qquad \mathbf{w_\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{1a}$$

$$\mathbf{y_t} = \mathbf{C}\mathbf{x_t} + \mathbf{v_t} = \mathbf{C}\mathbf{x_t} + \mathbf{v_\bullet}, \qquad \mathbf{v_\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{1b}$$

where $\mathbf{A} \in \mathcal{M}_{k\times k}$ (transition matrix) and $\mathbf{C} \in \mathcal{M}_{p\times k}$ (generative matrix). If we assume $\mathbf{x_1} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{1}, \mathbf{Q_1})$, then all future states $\mathbf{x_t}$ and observations $\mathbf{y_t}$ will also be gaussian distributed.

General Model

Several unsupervised methods can be represented as instances of an underlying basic model, such us: factor analysis, SPCA, PCA, ...

Basic generative model

$$\mathbf{x_{t+1}} = \mathbf{A}\mathbf{x_t} + \mathbf{w_t} = \mathbf{A}\mathbf{x_t} + \mathbf{w_\bullet}, \qquad \mathbf{w_\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{1a}$$

$$\mathbf{y_t} = \mathbf{C}\mathbf{x_t} + \mathbf{v_t} = \mathbf{C}\mathbf{x_t} + \mathbf{v_\bullet}, \qquad \mathbf{v_\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{1b}$$

where $\mathbf{A} \in \mathcal{M}_{k\times k}$ (transition matrix) and $\mathbf{C} \in \mathcal{M}_{p\times k}$ (generative matrix).
If we assume $\mathbf{x_1} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{1}, \mathbf{Q_1})$, then all future states $\mathbf{x_t}$ and observations $\mathbf{y_t}$ will also be gaussian distributed.
Notice: $P(\mathbf{x_{t+1}}|\mathbf{x_t}) = \mathcal{N}(\mathbf{A}\mathbf{x_t}, \mathbf{Q})|_{\mathbf{x_{t+1}}}$ and $P(\mathbf{y_t}|\mathbf{x_t}) = \mathcal{N}(\mathbf{C}\mathbf{x_t}|\mathbf{R})|_{\mathbf{y_t}}$, so we can calculate an explicit expression for the joint probability of a sequence of τ states and observables:
$P(\{\mathbf{x_1}, \dots, \mathbf{x_\tau}\}, \{\mathbf{y_1}, \dots, \mathbf{y_\tau}\}) = P(\mathbf{x_1}) \prod_{t=1}^{\tau-1} P(\mathbf{x_{t+1}}|\mathbf{x_t}) \prod_{t=1}^{\tau} P(\mathbf{y_t}|\mathbf{x_t})$.
Then, $-2\log P(\{\mathbf{x_1}, \dots, \mathbf{x_\tau}\}, \{\mathbf{y_1}, \dots, \mathbf{y_\tau}\})$ is the cost function used.

Inference and Learning

- **Inference or filtering and smoothing**: given a fixed model with parameters $\{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_1, \mathbf{Q}_1\}$, what can we say about the hidden states given some observations?

Inference and Learning

- **Inference or filtering and smoothing**: given a fixed model with parameters $\{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_1, \mathbf{Q}_1\}$, what can we say about the hidden states given some observations?
- **Learning (System identification)**: Given only an observed sequence $\{\mathbf{y}_1, \ldots, \mathbf{y}_\tau\}$ find the parameters $\{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_1, \mathbf{Q}_1\}$ that maximize the likelihood of the observed data.

Inference and Learning

- **Inference or filtering and smoothing**: given a fixed model with parameters $\{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_1, \mathbf{Q}_1\}$, what can we say about the hidden states given some observations?
- **Learning (System identification)**: Given only an observed sequence $\{\mathbf{y}_1, \ldots, \mathbf{y}_\tau\}$ find the parameters $\{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_1, \mathbf{Q}_1\}$ that maximize the likelihood of the observed data.

The basis is the **expectation-maximization (EM) algorithm**, whose goal is to maximize the likelihood of the observed data given the hidden states.

Inference and Learning

- **Inference or filtering and smoothing**: given a fixed model with parameters $\{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_1, \mathbf{Q}_1\}$, what can we say about the hidden states given some observations?
- **Learning (System identification)**: Given only an observed sequence $\{\mathbf{y}_1, \ldots, \mathbf{y}_\tau\}$ find the parameters $\{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_1, \mathbf{Q}_1\}$ that maximize the likelihood of the observed data.

The basis is the **expectation-maximization (EM) algorithm**, whose goal is to maximize the likelihood of the observed data given the hidden states. For any distribution $\mathcal{Q}$ over the hidden states variables, we can obtain a lower bound on $\mathcal{L}$:

$$\mathcal{L}(\theta) = \log P(\mathbf{Y}|\theta) \geq$$

$$\geq \int_{\mathbf{X}} \mathcal{Q}(\mathbf{X}) \log P(\mathbf{X}, \mathbf{Y}|\theta) d\mathbf{X} - \int_{\mathbf{X}} \mathcal{Q}(\mathbf{X}) \log \mathcal{Q}(\mathbf{X}) d\mathbf{X} := \mathcal{F}(\mathcal{Q}, \theta)$$

- **Inference or filtering and smoothing**: given a fixed model with parameters $\{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_1, \mathbf{Q}_1\}$, what can we say about the hidden states given some observations?
- **Learning (System identification)**: Given only an observed sequence $\{\mathbf{y}_1, \dots, \mathbf{y}_\tau\}$ find the parameters $\{\mathbf{A}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \boldsymbol{\mu}_1, \mathbf{Q}_1\}$ that maximize the likelihood of the observed data.

The basis is the **expectation-maximization (EM) algorithm**, whose goal is to maximize the likelihood of the observed data given the hidden states. For any distribution $\mathcal{Q}$ over the hidden states variables, we can obtain a lower bound on $\mathcal{L}$:

$$\mathcal{L}(\theta) = \log P(\mathbf{Y}|\theta) \geq$$

$$\geq \int_{\mathbf{X}} \mathcal{Q}(\mathbf{X}) \log P(\mathbf{X}, \mathbf{Y}|\theta) d\mathbf{X} - \int_{\mathbf{X}} \mathcal{Q}(\mathbf{X}) \log \mathcal{Q}(\mathbf{X}) d\mathbf{X} := \mathcal{F}(\mathcal{Q}, \theta)$$

E-step: $\mathcal{Q}_{k+1} \leftarrow \arg\max_{\mathcal{Q}} \mathcal{F}(\mathcal{Q}, \theta_k)$.

M-step: $\theta_{k+1} \leftarrow \arg\max_{\theta} \mathcal{F}(\mathcal{Q}_{k+1}, \theta)$.

Continuos-State Linear Gaussian Systems

From now on, the hidden state variable $\mathbf{x}$ is continuous and the noise processes are gaussian. First, we focus on static data models, i.e., data points are independently and identically distributed and they are not time dependent, so the equations (1) are simplified:

Continuos-State Linear Gaussian Systems

From now on, the hidden state variable $\mathbf{x}$ is continuous and the noise processes are gaussian. First, we focus on static data models, i.e., data points are independently and identically distributed and they are not time dependent, so the equations (1) are simplified:

$$\mathbf{x}_{\bullet} = \mathbf{w}_{\bullet}, \qquad \mathbf{w}_{\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{2a}$$

$$\mathbf{y}_{\bullet} = \mathbf{C}\mathbf{x}_{\bullet} + \mathbf{v}_{\bullet}, \qquad \mathbf{v}_{\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{2b}$$

Continuos-State Linear Gaussian Systems

From now on, the hidden state variable $\mathbf{x}$ is continuous and the noise processes are gaussian. First, we focus on static data models, i.e., data points are independently and identically distributed and they are not time dependent, so the equations (1) are simplified:

$$\mathbf{x}_\bullet = \mathbf{w}_\bullet, \qquad \mathbf{w}_\bullet \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{2a}$$

$$\mathbf{y}_\bullet = \mathbf{C}\mathbf{x}_\bullet + \mathbf{v}_\bullet, \qquad \mathbf{v}_\bullet \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{2b}$$

Inference: $P(\mathbf{x}_\bullet|\mathbf{y}_\bullet) = \mathcal{N}(\boldsymbol{\beta}\mathbf{y}_\bullet, I - \boldsymbol{\beta}C)|_{\mathbf{x}_\bullet}, \quad \boldsymbol{\beta} = \mathbf{C}^{\mathbf{T}}(\mathbf{C}\mathbf{C}^{\mathbf{T}} + \mathbf{R})^{-\mathbf{1}}$

Continuos-State Linear Gaussian Systems

From now on, the hidden state variable $\mathbf{x}$ is continuous and the noise processes are gaussian. First, we focus on static data models, i.e., data points are independently and identically distributed and they are not time dependent, so the equations (1) are simplified:

$$\mathbf{x}_\bullet = \mathbf{w}_\bullet, \qquad \mathbf{w}_\bullet \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{2a}$$

$$\mathbf{y}_\bullet = \mathbf{C}\mathbf{x}_\bullet + \mathbf{v}_\bullet, \qquad \mathbf{v}_\bullet \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{2b}$$

Inference: $P(\mathbf{x}_\bullet|\mathbf{y}_\bullet) = \mathcal{N}(\boldsymbol{\beta}\mathbf{y}_\bullet, I - \boldsymbol{\beta}C)|_{\mathbf{x}_\bullet}, \quad \boldsymbol{\beta} = \mathbf{C}^\mathbf{T}(\mathbf{C}\mathbf{C}^\mathbf{T} + \mathbf{R})^{-\mathbf{1}}$

- **Factor analysis**: $\mathbf{Q} = \mathbf{I}$, $\mathbf{R}$ is diagonal. Inference is done with the previous expression, and EM algorithm is used to learn $\mathbf{C}$ and $\mathbf{R}$.

Continuos-State Linear Gaussian Systems

From now on, the hidden state variable $\mathbf{x}$ is continuous and the noise processes are gaussian. First, we focus on static data models, i.e., data points are independently and identically distributed and they are not time dependent, so the equations (1) are simplified:

$$\mathbf{x_\bullet} = \mathbf{w_\bullet}, \qquad \mathbf{w_\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{2a}$$

$$\mathbf{y_\bullet} = \mathbf{C}\mathbf{x_\bullet} + \mathbf{v_\bullet}, \qquad \mathbf{v_\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{2b}$$

Inference: $P(\mathbf{x_\bullet}|\mathbf{y_\bullet}) = \mathcal{N}(\boldsymbol{\beta}\mathbf{y_\bullet}, I - \boldsymbol{\beta}C)|_{\mathbf{x_\bullet}}, \quad \boldsymbol{\beta} = \mathbf{C^T}(\mathbf{C}\mathbf{C^T} + \mathbf{R})^{-\mathbf{1}}$

- **Factor analysis**: $\mathbf{Q} = \mathbf{I}$, $\mathbf{R}$ is diagonal. Inference is done with the previous expression, and EM algorithm is used to learn $\mathbf{C}$ and $\mathbf{R}$.
- **SPCA**: $\mathbf{Q} = \mathbf{I}$, $\mathbf{R} = \alpha\mathbf{I}$. Inference and learning are like FA methods.

Continuos-State Linear Gaussian Systems

From now on, the hidden state variable $\mathbf{x}$ is continuous and the noise processes are gaussian. First, we focus on static data models, i.e., data points are independently and identically distributed and they are not time dependent, so the equations (1) are simplified:

$$\mathbf{x_\bullet} = \mathbf{w_\bullet}, \qquad \mathbf{w_\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{2a}$$

$$\mathbf{y_\bullet} = \mathbf{C}\mathbf{x_\bullet} + \mathbf{v_\bullet}, \qquad \mathbf{v_\bullet} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{2b}$$

Inference: $P(\mathbf{x_\bullet}|\mathbf{y_\bullet}) = \mathcal{N}(\boldsymbol{\beta}\mathbf{y_\bullet}, I - \boldsymbol{\beta}C)|_{\mathbf{x_\bullet}}, \quad \boldsymbol{\beta} = \mathbf{C^T}(\mathbf{CC^T} + \mathbf{R})^{-\mathbf{1}}$

- **Factor analysis**: $\mathbf{Q} = \mathbf{I}$, $\mathbf{R}$ is diagonal. Inference is done with the previous expression, and EM algorithm is used to learn $\mathbf{C}$ and $\mathbf{R}$.
- **SPCA**: $\mathbf{Q} = \mathbf{I}$, $\mathbf{R} = \alpha\mathbf{I}$. Inference and learning are like FA methods.
- **PCA**: Particular case of SPCA, where α tends to zero.

Continuos-State Linear Gaussian Systems

From now on, the hidden state variable $\mathbf{x}$ is continuous and the noise processes are gaussian. First, we focus on static data models, i.e., data points are independently and identically distributed and they are not time dependent, so the equations (1) are simplified:

$$\mathbf{x}_\bullet = \mathbf{w}_\bullet, \qquad \mathbf{w}_\bullet \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{2a}$$

$$\mathbf{y}_\bullet = \mathbf{C}\mathbf{x}_\bullet + \mathbf{v}_\bullet, \qquad \mathbf{v}_\bullet \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{2b}$$

Inference: $P(\mathbf{x}_\bullet|\mathbf{y}_\bullet) = \mathcal{N}(\boldsymbol{\beta}\mathbf{y}_\bullet, I - \boldsymbol{\beta}C)|_{\mathbf{x}_\bullet}, \quad \boldsymbol{\beta} = \mathbf{C}^{\mathbf{T}}(\mathbf{C}\mathbf{C}^{\mathbf{T}} + \mathbf{R})^{-1}$

- **Factor analysis**: $\mathbf{Q} = \mathbf{I}$, $\mathbf{R}$ is diagonal. Inference is done with the previous expression, and EM algorithm is used to learn $\mathbf{C}$ and $\mathbf{R}$.
- **SPCA**: $\mathbf{Q} = \mathbf{I}$, $\mathbf{R} = \alpha\mathbf{I}$. Inference and learning are like FA methods.
- **PCA**: Particular case of SPCA, where α tends to zero.
- **Kalman Filter Models**: Recovering equations (1) because of the time dependency (Linear dynamical systems). We can extend our spatial intuition of the static case to this dynamic model, but now, state-space ball "flows" from time step to time step.

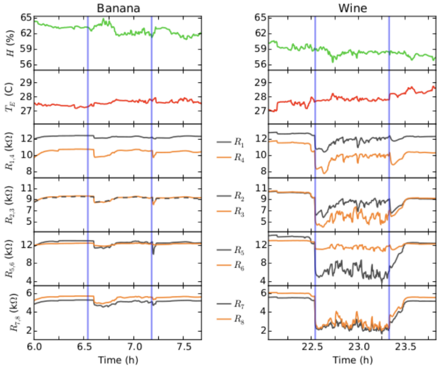
References (I)

- Sam Roweis and Zoubin Ghahramani
  A Unifying Review of Linear Gaussian Models.

Project description

Monitoring home activity with gas sensors

- Dataset available at UCI Machine Learning Repository
- The original article can be found at arxiv.org/pdf/1608.01719.pdf
- The main goals of the project are:
  - Detect stimulus
  - Classify stimulus

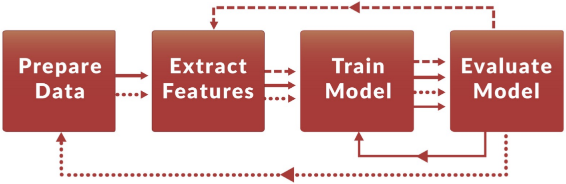


The metadata contains common information for each series

id	date	class	t0	dt
0	07-04-15	banana	13.49	1.64
1	07-05-15	wine	19.61	0.54
2	07-06-15	wine	19.99	0.66
3	07-09-15	banana	6.49	0.72
...				
98	09-16-15	background	14.41	0.71
99	09-17-15	background	11.93	0.68

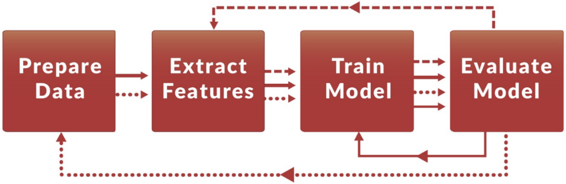
Iteration process

Common iteration over the different phases of a ML project



Iteration process

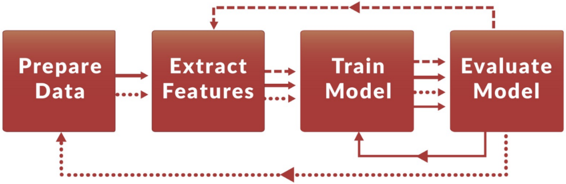
Common iteration over the different phases of a ML project



Datasets and features used:

- Raw data set
- Clean data set
- Dataset by windows: moving average

Common iteration over the different phases of a ML project



Supervised algorithms used:

- Logistic Regression
- Neural Networks
- Decision Trees
- Support Vector Machines
- Ensembles of the above
- Recurrent Neural Networks

Datasets and features used:

- Raw data set
- Clean data set
- Dataset by windows: moving average

Working with unbalanced data

- The original dataset has 80% examples of background, 11% of wine and 9% of banana readings. Accuracy turns out to be a poor score to validate the models. **F1-score** is more appropriate.

Working with unbalanced data

- The original dataset has 80% examples of background, 11% of wine and 9% of banana readings. Accuracy turns out to be a poor score to validate the models. **F1-score** is more appropriate.
- The models struggle to identify some minority class stimulus, so we execute **SMOTE**: Synthetic Minority Oversampling Technique.

Working with unbalanced data

- The original dataset has 80% examples of background, 11% of wine and 9% of banana readings. Accuracy turns out to be a poor score to validate the models. **F1-score** is more appropriate.
- The models struggle to identify some minority class stimulus, so we execute **SMOTE**: Synthetic Minority Oversampling Technique.

Best results

Acc.—F1-score	**Raw DB**	**Clean DB**	**Win DB**	**SMOTE DB**
Ensembles NN 3x4 - 0.01	85%—78%	85%—80%	86%—84%	85%—85%
Random Forest	84%—79%	84%—81%	86%—83%	85%—84%
Original paper (SVM)	77%—?	?—?	81%—?	?—?

References (II)

- Online Decorrelation of Humidity and Temperature in Chemical Sensors for Continuous Monitoring

  Ramon Huerta ,Thiago Mosqueiro, Jordi Fonollosa, Nikolai F. Rulkov and Irene Rodriguez-Lujan

- Gas sensors for home activity monitoring
  Machine Learning Repository
  Flavia Huerta, Ramon Huerta

- Smote oversampling for imbalanced classification
  Machine Learning Mastery.
  Jason Brownlee