Fondements de l'apprentissage machine

Automne 2014

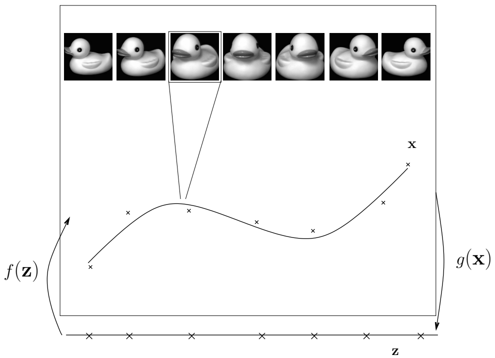
Roland Memisevic

Leçon 7

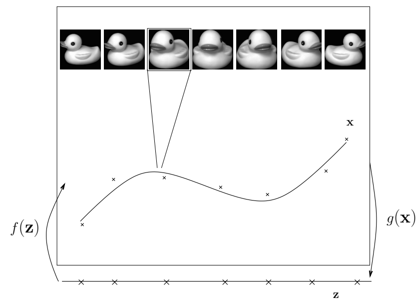
Plan

- Variables latentes continues
- Principal Components Analysis (PCA)
- Probabilistic PCA (PPCA)
- Variables latentes continues non-linéaires

Variables latentes continues et réduction de dimension

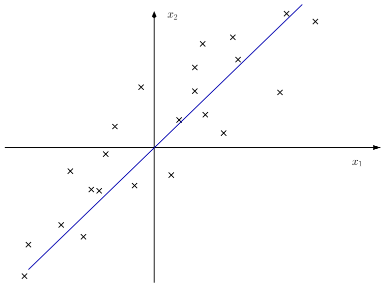


Variables latentes continues et réduction de dimension



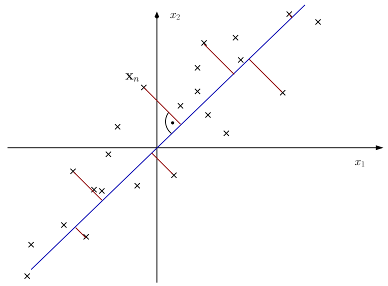
- Dans certains applications, en plus de représentants $\mathbf{z}$ pour les données d'entraînement, on cherche aussi
  1. Une **fonction backward** $g(\mathbf{x})$ qui peut être appliquée à des données de test.
  2. Une **fonction forward** $f(\mathbf{z})$ qui imagine de nouvelles données.

Principal Components Analysis (PCA)



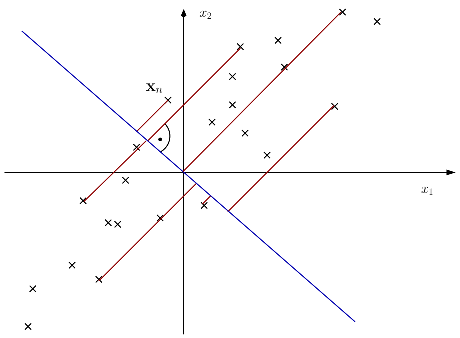
- Si une variété (manifold) est *linéaire* (donc un sous-espace), l'apprentissage est simple et peut être fait sous forme fermée.
- Le problème de trouver ce sous-espace s'appelle Principal Components Analysis (PCA) (ou Analyse en Composantes Principales, ACP)
- L'inférence correspond à une projection sur le sous-espace.

Minimiser les erreurs de projection ou maximiser la variance



- La variance des projections est **grande**.
- L'erreur de projection est **faible**.

Minimiser les erreurs de projection ou maximiser la variance



- La variance des projections est **faible**.
- L'erreur de projection est **grande**.

Principal Components Analysis

- Pour apprendre un *sous-espace*, nous devons d'abord centrer les données :

$$\frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n = \mathbf{0}$$

- Pour dériver la solution de PCA, nous définissons une *base orthonormale* pour le sous-espace consistant de vecteurs

$$\mathbf{u}_1, \ldots, \mathbf{u}_M$$

  où M est plus petit que la dimensionalité des données.
- PCA essaie d'apprendre cette base.

Principal Components Analysis

- Il serait pratique d'empiler les vecteurs de base côte à côte dans une matrice $\mathbf{U}$.
- Supposons nous avons déja estimé la base optimale. Dans ce cas, nous pouvons écrire les fonctions *forward* et *backward* comme suit :

Fonction backward

- Les coefficients optimaux pour approximer $\mathbf{x}$ dans le sous-espace sont donnés par

$$\mathbf{z} = \mathbf{U}^{\mathrm{T}}\mathbf{x}$$

Fonction forward

- L'approximation $\tilde{\mathbf{x}}$ de $\mathbf{x}$ est

$$\tilde{\mathbf{x}} = \mathbf{U}\mathbf{z} = \mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{x}$$

Principal Components Analysis

- Pour apprendre le sous-espace : minimisez l'erreur de reconstruction :

$$E(\mathbf{U}) = \sum_n \|\mathbf{x}_n - \mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{x}_n\|^2$$

sous la contrainte $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$

- Solution : Empilez les données rangée par rangée dans la matrice $\mathbf{X}$ et écrivez l'objectif comme *forme quadratique* de $\mathbf{U}$:

$$\begin{aligned} E(\mathbf{U}) &= \|\mathbf{X}^{\mathrm{T}} - \mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\|_F^2 \\ &= \mathrm{Tr}((\mathbf{X}^{\mathrm{T}} - \mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}})^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}} - \mathbf{U}\mathbf{U}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}})) \\ &= \mathrm{Tr}(\mathbf{X}\mathbf{X}^{\mathrm{T}}) - \mathrm{Tr}(\mathbf{U}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{U}) \\ &= -\mathrm{Tr}(\mathbf{U}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{U}) + \mathrm{const} \end{aligned}$$

Principal Components Analysis

Optimisation des formes quadratiques

- La matrice $\mathbf{U}$ qui maximise

$$\mathrm{Tr}(\mathbf{U}^{\mathrm{T}}\mathbf{A}\mathbf{U})$$

soumise à la contrainte

$$\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$$

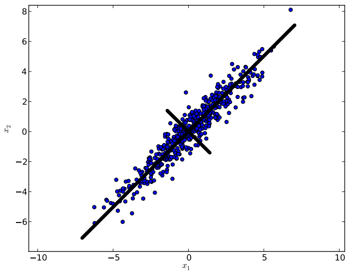
est la matrice dont les colonnes sont les vecteurs propres de $\mathbf{A}$ correspondant aux M plus grandes valeurs propres.

Principal Components Analysis

Algorithme PCA

1. Centrez les données.
2. Calculez la matrice de covariance $\mathbf{C} = \frac{1}{N}\mathbf{X}^{\mathrm{T}}\mathbf{X}$.
3. Effectuez une décomposition de $\mathbf{C}$ en vecteurs propres.
4. Triez les vecteurs propres en fonction de la taille de leurs valeurs propres.
5. Empilez les M vecteurs propres principaux dans la matrice $\mathbf{U}$.

Principal Components Analysis



- Des données en deux dimensions et leurs deux composantes principales.
- Les projections conservent la majorité de la variabilité des données. Alors PCA peut effectuer une compression avec perte.
- D'autres applications : visualisation, pré-traitement pour apprentissage supervisé, etc.

PCA et whitening (blanchiment)

- Les composantes de la représentation latente, $\mathbf{Z}$, sont décorrélées ($\mathbf{Z}$ a une matrice de covariance diagonale) :

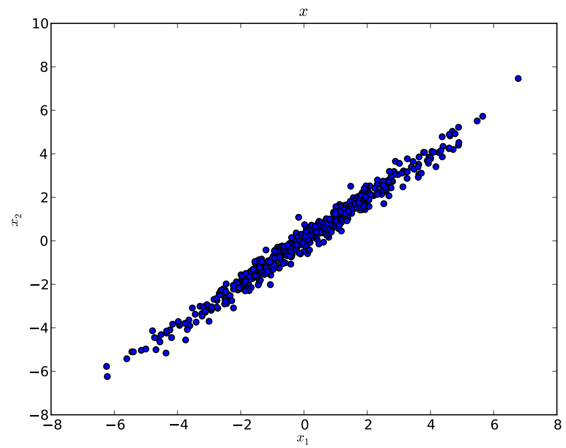
$$\begin{aligned}\frac{1}{N}\sum_n \mathbf{z}_n\mathbf{z}_n^{\mathrm{T}} &= \frac{1}{N}\sum_n \mathbf{U}^{\mathrm{T}}\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\mathbf{U}\\ &= \mathbf{U}^{\mathrm{T}}\Big(\frac{1}{N}\sum_n \mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\Big)\mathbf{U}\\ &= \mathbf{U}^{\mathrm{T}}\mathbf{C}\mathbf{U}\\ &= \mathbf{L}\end{aligned}$$

  où la matrice diagonale $\mathbf{L}$ contient les valeurs propres de $\mathbf{C}$ sur sa diagonale.
- (La dernière étape découle de la définition des valeurs propres : $\mathbf{C}\mathbf{u}_i = \lambda_i\mathbf{u}_i$)

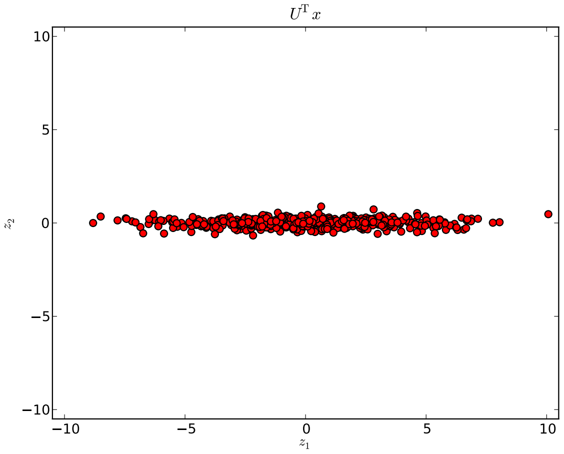
PCA et whitening (blanchiment)

- Nous pouvons obtenir l'*identité* comme matrice de covariance pour $\mathbf{Z}$ si, à la place de $\mathbf{U}^{\mathrm{T}}$, nous prenons $\mathbf{L}^{-\frac{1}{2}}\mathbf{U}^{\mathrm{T}}$ comme fonction forward.
- Des données avec une matrice de covariance identité sont appelées white, et la multiplication par $\mathbf{L}^{-\frac{1}{2}}\mathbf{U}^{\mathrm{T}}$ whitening.
- On peut effectuer le whitening sans réduction de la dimensionalité, cad. en utilisant $M = D$.

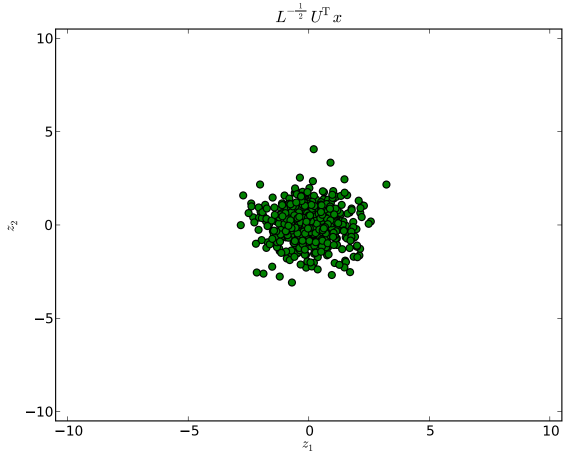
Whitening : exemple



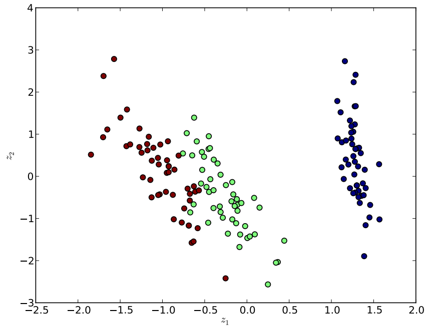
Whitening : exemple



Whitening : exemple



PCA pour la visualization



- Projection de base de données "Iris" en deux dimensions (4 dimensions originellement).

Probabilistic PCA (PPCA)

- On peut définir PCA comme un modèle de variables latentes probabilistes.
- Supposez une distribution a priori gaussienne :

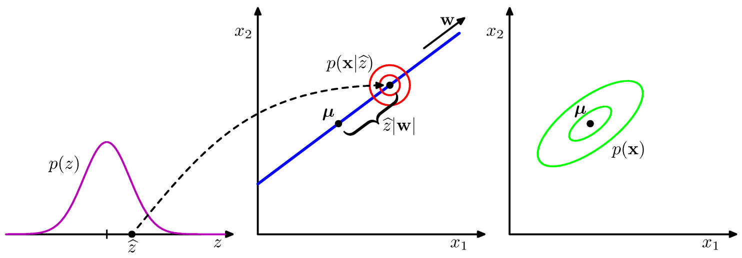
$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$$

sur les variables latentes, ainsi qu'une distribution conditionelle gaussienne :

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \sigma^2\mathbf{I})$$

sur les observations.

Probabilistic PCA (PPCA)



- Cela définit un processus génératif, où nous tirons d'abord un vecteur d'une distribution gaussienne de dimension M dans l'espace latent, puis nous tirons une observation d'une gaussienne de dimension D dont la moyenne dépend du vecteur latent.

Probabilistic PCA (PPCA)

- Pour la fonction backward, utilisez la règle de Bayes :

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{\int_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z}}$$

pour obtenir :

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu}), \sigma^{-2}\mathbf{M}) \text{ avec } \mathbf{M} = \mathbf{W}^{\mathrm{T}}\mathbf{W}+\sigma^2\mathbf{I}$$

- La vraisemblance est une Gaussienne avec une covariance sous contrainte :

$$p(\mathbf{x}) = \int_{\mathbf{z}} p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\ \mathrm{d}\mathbf{z} = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^{\mathrm{T}} + \sigma^2\mathbf{I})$$

- Pour l'optimiser, utilisez soit la descente de gradient ou EM.
- (voir Bishop, 2.3.3)

EM pour PPCA

- E-step : calculez $p(\mathbf{z}_n|\mathbf{x}_n)$
- M-step : maximisez l'expected complete log-likelihood.
- Comme les distributions a posteriori, $p(\mathbf{z}_n|\mathbf{x}_n)$, sont des gaussiennes, elles sont représentées par leur moyenne et covariance.
- (voir Bishop, page 578)
- Des avantages de la formulation probabiliste de PCA :
  - (i) il est facile de traiter des données manquantes
  - (ii) il est facile de définir un mélange de modèles PCA (clustering et réduction de la dimensionalité en même temps)
  - (iii) il existe une formulation bayésienne complète où les parametres sont marginalisés.

Analyse de facteurs (Factor Analysis)

- L'Analyse de facteurs correspond à un modèle génératif très semblable à PCA probabiliste.
- La seule différence est que le bruit gaussien final ajouté est diagonal plutôt que sphérique

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Psi})$$

avec $\boldsymbol{\Psi}$ diagonal.

- L'apprentissage se fait comme pour PPCA.

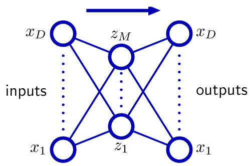
Voir diapo par Pascal Vincent ift3395 2013

Comment choisir le nombre de composantes ?

- Soit en se basant sur la log-vraisemblance obtenue sur un ensemble de validation.
- Soit en adoptant une approche bayésienne en précisant des distributions à priori.

Voir diapo par Pascal Vincent ift3395 2013

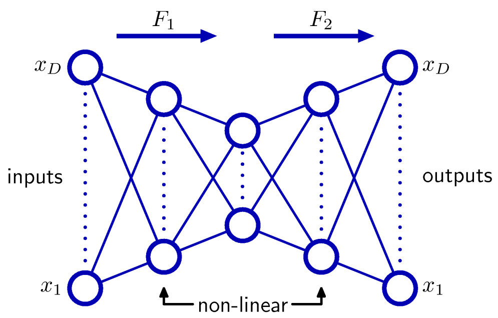
Réseaux de neurones auto-associateurs (auto-encoder)



- Un réseau de neurones de type feed-forward est entraîné à reproduire son entrée (cible=observations).
- La couche cachée est choisie de dimension $M < D$. Ceci entraîne des erreurs de reconstruction, que l'entraînement cherche à minimiser.
- On obtient une représentation de dimension réduite au niveau de la couche cachée.
- Si le réseau est linéaire, c'est équivalent à PCA.

Voir diapo par Pascal Vincent ift3395 2013

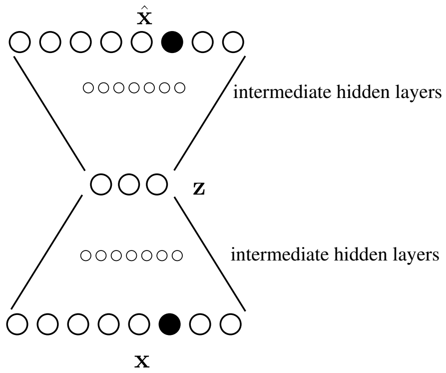
Réseaux de neurones auto-associateurs (auto-encoder)



- S'il y a plusieurs couches cachées avec des non-linéarités, c'est une méthode de réduction de dimensionalité non-linéaire.

Réseaux de neurones auto-associateurs (auto-encoder)

- Un problème jouet classique :



Modélisation de variétés non-linéaires (manifold learning)

- On peut modéliser une variété non-linéaire par un mélange de modèles linéaires, par ex. un mélange d'analyses de facteurs.
- Autres méthodes de réduction de dimensionalité non-linéaires :
  - Multidimensional Scaling (MDS)
  - Locally Linear Embedding (LLE)
  - Isometric Feature Mapping (Isomap)
  - ...

Voir diapo par Pascal Vincent ift3395 2013

Analyse en composantes indépendantes (ICA)

- Similaire à PCA probabiliste, mais la distribution sur les variables latentes (les composantes) n'est pas gaussienne, mais factorielle, ce qui correspond à avoir des composantes *indépendantes* :

$$p(\mathbf{z}) = \prod_{j=1}^{M} p(z_j)$$

Voir diapo par Pascal Vincent ift3395 2013