Terminology

(data) can be : bare ~~xxx~~ numbers
: symbols
: ~~xxxxxx~~ visual (images)
: audio files
: text .

(Pre Processing)

(we assume there is structure)

- Filtering (noise reduction) find ~~x~~ how)
- Normalization.
- Outlier ~~detecn~~ detection.
- dimensionality ~~xxxxxx~~ reduction

(euclidean distance fails if dimension is high, say 10,000)

(Features) / Attributes
- data representation

hand Crafted / engineered.
Eg. SIFT.

Automatically ~~extract~~ extracted features
eg ~~SIFT.~~ CNN.

(Encoding)

- compressing data (purpose is not dimensionality reduct^n, purpose is encoding)

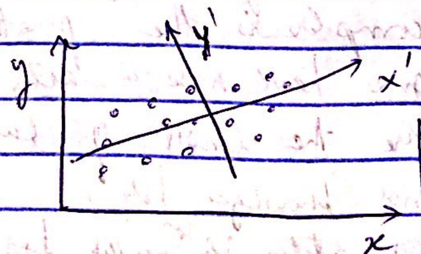
~~Conventional~~ Conventional (PCA, Fisher Vector)

ML (autoencoders, t-SNE)

Jan 10th 2016

Data → [Compression Encoding] → MI/ML [Learning Classification optimization Approximation] → Data → [Compress / Encode .]

- Compression PCA
- Encoding (Fisher Vector)



variance of data maximum when projected on x'.

Data ~~that~~ does not give me variance is useless! It shud give change information

PCA finds x'/y' with max. variance wrt. population mean.
PCA repeats the process unit N such axes are found.

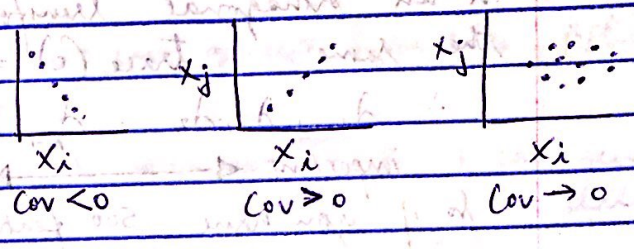
PCA finds a set of orthogonal axes that diagonalizes the covariance matrix.

audio

$x \longrightarrow = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

one datapoint

features that describe x.



Cov. Matrix $= \sum_{ij} = \text{cov}(x_i, x_j) = E\left[(x_i - \mu_i)(x_j - \mu_j)\right]$

$\mu_i = E(x_i)$

$\mu_j = E(x_j)$

Generalization of variance:

Covariance $\Sigma = E\left[(X - E(X))(X - E(X))^T\right]$

$\sigma^2 = \text{var}(x) = E\left[(X - E(X))^2\right]$

Entries of diagonal of Σ are the variances of each element of X.

x = scalar X = set

$\mathbf{x}$ = vector $\mathbf{X}$ = matrix

Covariance $C = E\left[(x_p - m)(x_p - m)^T\right]$

x_p = sample/population, m = mean

Expected value.

We can estimate C as: $C = \frac{1}{p} \sum_{p=1}^{p} x_p x_p^T - m m^T$

$m = \frac{1}{p} \sum_{p=1}^{p} x_p$

Diagonalizing C using a suitable orthogonal transformation matrix A by obtaining N orthogonal eigen vectors u_i with eigen value λ_i.

$C u_i = \lambda_i u_i$, $i = 1, 2 \ldots N$.

The vectors u_i are derived $u_i = A(x_i - m)$ $x_i = m + A^T u_i$

Since $A^{-1} = A^T$ (because its orthogonal)

Hence $C' = ACA^T$

$$C' = \begin{bmatrix} \lambda_1 & & 0 \\ & \lambda_2 \ldots & \\ 0 & & \lambda_N \end{bmatrix}$$

In an orthogonal transformation the "trace" of a matrix remains the same. $\text{trace}(C) = \text{trace}(C') = \sum_{i=1}^{N} \lambda_i = \sum_{i=1}^{N} \delta_i^2$

$\therefore \lambda_1, \lambda_2, \lambda_3 \ldots \lambda_{N-1}, \lambda_N$

important ⟵⟶ useless

So if you have 500 features, If we can reconstruct entire data using only top 2, with least error.

<u>Encoding</u>

<u>eg.</u> Fisher Vector.

Applies Fisher kernel. $X = \{ x_t : t = 1, 2 \ldots T \}$

T observations ~~$x_t \in T$~~

$x_t \in X$

(λ = vector)

μ_λ (parameter) probability density function which models the generative process of elements in ~~X~~ X

$\lambda = [\lambda_1, \lambda_2 \ldots \lambda_M] \in \mathbb{R}^M$

M # of parameters of μ_λ

(Statistics) Score func^n: this is the gradient (partial derivatives) wrt the parameters λ of the logarithm (natural log) of the liklihood function (log. liklihood)

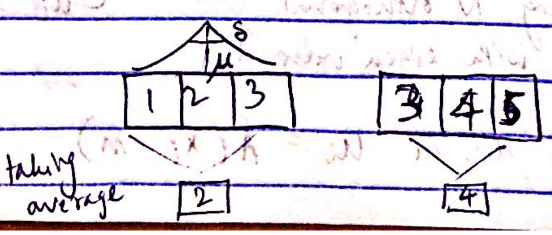
Score function: $G_\lambda^X = \nabla_\lambda \log \mu_\lambda(x)$

$= \nabla_\lambda \log P(X/\mu_\lambda)$

The gradient quantifies the contribution of individual parameters λ

→ How to change λ (of μ_λ) to better fit data (x)?

parameters $\begin{cases} \lambda_1 = \mu \\ \lambda_2 = \delta \end{cases}$



PCA = compression.
Fisher .. = encoding

Since $\nabla_\lambda^x \in \mathbb{R}^M$ ⇒ it does not depend on T (size of data) (depends on M, not T)

where $M << T$.

$X = \{x_t : t = 1, 2 \dots T\}$

parameters of probability density function.

(representation) (data)

This is encoding, not compression. we can generate new data. After fitting, we have generative data..

Encoding because after I find M ~~dimensions~~ parameters, I can generate data

From information geometry, $U = \{\mu_\lambda : \lambda \in \Lambda\}$

is a Riemannian manifold M_Λ with a local metric,

given by fisher Information Matrix F_λ

topology space with local euclidean metric.

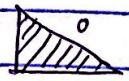
(manifold: hyperdimensional space locally, euclidean distance (can be used, not globally)

$$F_\lambda = E_{x-\mu_\lambda}\left[\nabla_\lambda^x \nabla_\lambda^{xT}\right]$$

Using Fisher Kernel K_{FK} to measure the similarity between two samples x and y.

$$K_{FK}(x,y) = \nabla_\lambda^{xT} F_\lambda^{-1} \nabla_\lambda^x$$

Using Cholesky decomposition: $F_\lambda^{-1} = L_\lambda^T L_\lambda$ → lower triangular matrix



We can re-write K_{FK}

$$K_{FK}(x,y) = g_\lambda^{xT} g_\lambda^x$$

with $g_\lambda^x = L_\lambda \nabla_\lambda^x$

$= L_\lambda \nabla_\lambda \log P(x/\mu_\lambda)$

Fisher vector