Advanced Machine Learning from Theory to Practice Lecture 4 Dimension Reduction and Feature Design

F. d'Alche-Buc and E. Le Pennec

Fall 2015

Outline

1. Dimension Reduction
    - Reconstruction error
    - Distance preservation
2. Feature Design
    - Renormalization
    - Basis and Dictionary Learning
    - Categorical Feature Encoding
    - Quantization and Binarization
    - Hashing
    - Pooling
    - Application Specific Features

- Training data : $\mathcal{D} = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \in \mathcal{X}^n$ (i.i.d. $\sim \mathbf{P}$)
- Space $\mathcal{X}$ of possibly high dimension.

Dimension Reduction Map

- Construct a map Φ from the space $\mathcal{X}$ into a space $\mathcal{X}'$ of smaller dimension :

$$\begin{aligned} \Phi : \quad & \mathcal{X} \to \mathcal{X}' \\ & \mathbf{X} \mapsto \Phi(\mathbf{X}) \end{aligned}$$

Criterion

- Reconstruction error
- Distance preservation

Goal

- Construct a map Φ from the space $\mathcal{X}$ into a space $\mathcal{X}'$ of smaller dimension :

$$\begin{aligned} \Phi : \quad & \mathcal{X} \to \mathcal{X}' \\ & \mathbf{X} \mapsto \Phi(\mathbf{X}) \end{aligned}$$

- Construct $\widetilde{\Phi}$ from $\mathcal{X}'$ to $\mathcal{X}$
- Control the error between $\mathbf{X}$ and its reconstruction $\widetilde{\Phi}(\Phi(\mathbf{X}))$

- Canonical example for $\mathbf{X} \in \mathbb{R}^d$: find Φ and $\widetilde{\Phi}$ in a parametric family that minimize

$$\frac{1}{n}\sum_{i=1}^{n} \|\mathbf{X}_i - \widetilde{\Phi}(\Phi(\mathbf{X}_i))\|^2$$

- $\mathcal{X} \in \mathbb{R}^d$ and $\mathcal{X}' = \mathbb{R}^{d'}$
- Affine model $\mathbf{X} \sim m + \sum_{l=1}^{d'} \mathbf{X}'_l V^{(l)}$ with $(V^{(l)})$ an orthonormal basis.
- Equivalent to :

$$\Phi(\mathbf{X}) = V^t(\mathbf{X} - m) \quad \text{and} \quad \widetilde{\Phi}(\mathbf{X}') = m + V\mathbf{X}'$$

- Reconstruction error criterion :

$$\frac{1}{n}\sum_{i=1}^{n} \|\mathbf{X}_i - (m + VV^t(\mathbf{X}_i - m)\|^2$$

- **Explicit solution :** m is the empirical mean and V is any orthonormal basis of the space spanned by the d' first eigenvectors (the one with largest eigenvalues) of the empirical covariance matrix $\frac{1}{n}\sum_{i=1}^{n}(\mathbf{X}_i - m)(\mathbf{X}_i - m)^t$.

PCA Algorithm

- Compute the empirical mean $m = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i$
- Compute the empirical covariance matrix $\frac{1}{n}\sum_{i=1}^{n}(\mathbf{X}_i - m)(\mathbf{X}_i - m)^t$.
- Compute the d' first eigenvectors of this matrix : $V^{(1)}, \ldots, V^{(d')}$
- Set $\Phi(\mathbf{X}) = V^t(\mathbf{X} - m)$

- Complexity : $O(n(1 + d^2) + d'd^2)$
- Interpretation :
  - $\Phi(\mathbf{X}) = V^t(\mathbf{X} - m)$: coordinates in the restricted space.
  - $V^{(i)}$: influence of each original coordinates in the ith new one.
- **Scaling :** This method is not invariant to a scaling of the variables ! It is custom to normalize the variables (at least within groups) before applying PCA.

- PCA assumes $\mathcal{X} = \mathbb{R}^d$!
- How to deal with categorical values ?
- MFA = PCA with clever coding strategy for categorical values.

Categorical value code for a single variable

- Classical redundant dummy coding :

$$\mathbf{X} \in \{1, \ldots, V\} \mapsto P(\mathbf{X}) = (\mathbf{1}_{\mathbf{X}=1}, \ldots, \mathbf{1}_{\mathbf{X}=V})^t$$

- Compute the mean (i.e. the empirical proportion) $\overline{P} = \frac{1}{n} P(\mathbf{X})$
- *Renormalize* $P(\mathbf{X})$ by $1/\sqrt{\overline{P}}$:

$$P(\mathbf{X}) = (\mathbf{1}_{\mathbf{X}=1}, \ldots \mathbf{1}_{\mathbf{X}=V}) \mapsto P^r(\mathbf{X}) = \left(\frac{\mathbf{1}_{\mathbf{X}=1}}{\sqrt{\overline{P}_1}}, \ldots, \frac{\mathbf{1}_{\mathbf{X}=V}}{\sqrt{\overline{P}_V}} \right)$$

- χ^2 type distance !

- PCA becomes the minimization of

$$\frac{1}{n}\sum_{i=1}^{n}\|P^r(\mathbf{X}_i)-(m+VV^t(P^r(\mathbf{X}_i)-m))\|^2$$

$$=\frac{1}{n}\sum_{i=1}^{n}\sum_{v=1}^{V}\left|\frac{\mathbf{1}_{\mathbf{X}_i=v}-(m'+\sum_{l=1}^{d'}V^{(l)t}(P(\mathbf{X}_i)-m')V^{(l,v)})}{\overline{P}_v}\right|^2$$

- Interpretation :
  - $m'=\overline{P}$
  - $\Phi(\mathbf{X})=V^t(P^r\mathbf{X}-m)$: coordinates in the restricted space.
  - $V^{(l)}$ appears as a probability profile.
- Complexity : $O(n(1+V^2)+d'V^2)$
- Link with Correspondence Analysis (CA)

MFA Algorithm

- Redundant dummy coding of each categorical variable.
- Renormalization of each block of dummy variable.
- Classical PCA algorithm on the resulting variables

- Interpretation as a reconstruction error with a rescaled/χ^2 metric.
- Interpretation :
  - $\Phi(\mathbf{X}) = V^t(P^r(\mathbf{X}) - m)$: coordinates in the restricted space.
  - $V^{(i)}$: influence of each modality/variable in the ith new coordinates.
- **Scaling :** This method is not invariant to a scaling of the continuous variables ! It is custom to normalize the variables (at least within groups) before applying PCA.

PCA Model

- PCA : Linear model assumption

$$\mathbf{X} \simeq m + \sum_{l=1}^{d'} \mathbf{X}'_l V^{(l)}$$

- with
  - $V^{(l)}$ orthonormal
  - $\mathbf{X}'_l$ without constrains.

- Two directions of extension :
  - Other constrains on V (or the coordinates in the restricted space) : ICA, NMF, Dictionary approach
  - PCA on a non linear image of $\mathbf{X}$: kernel-PCA
- Much more complex algorithm !

ICA (Independent Component Analysis)

- Linear model assumption

$$\mathbf{X} \simeq m + \sum_{l=1}^{d'} \mathbf{X}'_l V^{(l)}$$

- with
  - $V^{(l)}$ without constrains.
  - $\mathbf{X}'_l$ independent

NMF (Non Negative Matrix Factorization)

- (Linear) Model assumption

$$\mathbf{X} \simeq m + \sum_{l=1}^{d'} \mathbf{X}'^{(l)} V^{(l)}$$

- with
  - $V^{(l)}$ non negative
  - $\mathbf{X}'_l$ non negative.

Dictionary

- (Linear) Model assumption

$$\mathbf{X} \simeq m + \sum_{l=1}^{d'} \mathbf{X}'_l V^{(l)}$$

- with
  - $V^{(l)}$ without constrains
  - $\mathbf{X}'$ sparse (with a lot of 0)

kernel PCA

- Linear model assumption

$$\Psi(\mathbf{X}) \simeq m + \sum_{l=1}^{d'} \mathbf{X}'_l V^{(l)}$$

- with
  - $V^{(l)}$ orthonormal
  - $\mathbf{X}'_l$ without constrains.

- Linear model assumption :

$$\mathbf{X} \simeq m + \sum_{l=1}^{d'} \mathbf{X}'_l V^{(l)}$$

- Vector rewriting

$$\mathbf{X}^t \simeq m^t + \mathbf{X}'^{,t} V^t$$

- Matrix rewriting

$$\begin{pmatrix} \mathbf{X}_1^t - m^t \\ \vdots \\ \mathbf{X}_n^t - m^t \end{pmatrix} \simeq \begin{pmatrix} \mathbf{X}_1'^t \\ \vdots \\ \mathbf{X}_n'^t \end{pmatrix} V^t$$

- Low rank matrix factorization !
- Truncated SVD solution

- Different point of view!
- Focus on pairwise distance $d(\mathbf{X}_i, \mathbf{X}_j)$.

Distance Preservation

- Construct a map Φ from the space $\mathcal{X}$ into a space $\mathcal{X}'$ of smaller dimension :

$$\begin{aligned}\Phi : \quad & \mathcal{X} \to \mathcal{X}' \\ & \mathbf{X} \mapsto \Phi(\mathbf{X}) = \mathbf{X}'\end{aligned}$$

- such that

$$d(\mathbf{X}_i, \mathbf{X}_j) \sim d'(\mathbf{X}'_i, \mathbf{X}'_j)$$

- Most natural criterion :

$$\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left| d(\mathbf{X}_i, \mathbf{X}_j) - d'(\mathbf{X}'_i, \mathbf{X}'_j) \right|^2$$

- Φ often defined only on $\mathbf{D}$...

Random Projection Heuristic

- Draw at random d' unit vector (direction) U_i.
- Use $\mathbf{X}' = U^t(\mathbf{X} - m)$ with $m = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i$

- **Property :** If $\mathbf{X}$ lives in a space of dimension d'', then, as soon as, $d' \sim d'' \log(d'')$,

$$\|\mathbf{X}_i - \mathbf{X}_j\|^2 \sim \frac{d}{d'}\|\mathbf{X}'_i - \mathbf{X}'_j\|^2$$

- Do not really use the data !

LLE Heuristic

- For each point $\mathbf{X}_i$, define a neighborhood $\mathcal{N}_i$ (either by a distance or a number of points).
- Compute some weights $W_{i,j}$ such that

$$W_{i,j} = 0 \quad \text{if } \mathbf{X}_j \notin \mathcal{N}_i$$
$$\mathbf{X}_i \sim \sum_j W_{i,j} \mathbf{X}_j$$

- Find some $\mathbf{X}'_i$ in a space $\mathcal{X}'$ of smaller dimension such that

$$\mathbf{X}'_i \sim \sum_j W_{i,j} \mathbf{X}'_j$$

- LLE : use a least square metric for the fits.

MDS Heuristic

- If $d(x,y) = \|x-y\|^2$, one can compute a Gram matrix
$$(\mathbf{X}_i - m)^t(\mathbf{X}_j - m)$$
for $m = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X}_i$
- Match the *scalar* products :
$$\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|(\mathbf{X}_i - m)^t(\mathbf{X}_j - m) - \mathbf{X}_i'^t\mathbf{X}_j'\right|^2$$
- Linear method : $\mathbf{X}' = U^t(\mathbf{X} - m)$ with U orthonormal

- **Beware : X** is unknown !

- Resulting criterion : minimization in $U^t(\mathbf{X}_i - m)$ of

$$\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left|(\mathbf{X}_i - m)^t(\mathbf{X}_j - m) - (\mathbf{X}_i - m)^t UU^t(\mathbf{X}_j - m)\right|^2$$

  without knowing explicitly $\mathbf{X}$...
- Explicit solution obtained through the eigendecomposition of the know Gram matrix $(\mathbf{X}_i - m)^t(\mathbf{X}_j - m)$ by keeping only the d' largest eigenvalues.
- In this case, MDS yields the same result than the PCA (but with different inputs, distance between observation vs correlations) !

- **Explanation :** Same SVD problem up to a transposition :
  - MDS
$$\overline{\mathbf{X}}_{(n)}^t \overline{\mathbf{X}}_{(n)} \sim \overline{\mathbf{X}}_{(n)}^t U U^t \overline{\mathbf{X}}_{(n)}$$
  - PCA
$$\overline{\mathbf{X}}_{(n)} \overline{\mathbf{X}}_{(n)}^t \sim U^t \overline{\mathbf{X}}_{(n)} \overline{\mathbf{X}}_{(n)}^t U$$
- Complexity : ACP $O(d' d^2)$ vs MDS $O(d' n^2)$...

MDS

- Apply this algorithm even if $d(x, y) \neq \|x - y\|^2$!

- **True distance minimization :** Simple gradient descent can be used (can be stuck in local minima).

- MDS : equivalent to PCA (but more expensive) if $d(x, y) = \|x - y\|^2$!
- ISOMAP : use a *localized* distance instead to limit the influence of very far point.

ISOMAP

- For each point $\mathbf{X}_i$, define a neighborhood $\mathcal{N}_i$ (either by a distance or a number of points) and let

$$d(\mathbf{X}_i, \mathbf{X}_j) = \begin{cases} 0 & \text{if } \mathbf{X}_j \notin \mathcal{N}_i \\ \|\mathbf{X}_i - \mathbf{X}_j\|^2 & \text{otherwise} \end{cases}$$

- Use the MDS algorithm with this modified distance

Graph heuristic

- Construct a graph with weighted edges $w_{i,j}$ measuring the *proximity* of $\mathbf{X}_i$ and $\mathbf{X}_j$ ($w_{i,j}$ large if close and 0 if there is no information).
- Find the points $\mathbf{X}'_i \in \mathbb{R}^{d'}$ minimizing

$$\frac{1}{n}\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{i,j}\|\mathbf{X}'_i - \mathbf{X}'_j\|^2$$

- Need of a constraint on the size of $\mathbf{X}'_i$...
- Explicit solution through linear algebra : d' eigenvectors with smallest eigenvalues of the Laplacian of the graph $D - W$, where D is a diagonal matrix with $D_{i,i} = \sum_j w_{i,j}$.
- Variation on the definition of the Laplacian...

SNE heuristic

- From $\mathbf{X}_i \in \mathcal{X}$, construct a set of conditional probability :
$$P_{j|i} = \frac{e^{-\|\mathbf{X}_i - \mathbf{X}_j\|^2/2\sigma_i^2}}{\sum_{k\neq i} e^{-\|\mathbf{X}_i - \mathbf{X}_j\|^2/2\sigma_i^2}} \qquad P_{i|i} = 0$$
- Find $\mathbf{X}'_i$ in $\mathbb{R}^{d'}$ such that the set of conditional probability :
$$Q_{j|i} = \frac{e^{-\|\mathbf{X}'_i - \mathbf{X}'_j\|^2/2\sigma_i^2}}{\sum_{k\neq i} e^{-\|\mathbf{X}'_i - \mathbf{X}'_j\|^2/2\sigma_i^2}} \qquad Q_{i|i} = 0$$

  is close from P.

- **t-SNE :** use a Student-t term $(1 + \|\mathbf{X}'_i - \mathbf{X}'_j\|^2)^{-1}$ for $\mathbf{X}'_i$
- Minimize the Kullback-Leibler divergence $(\sum_{i,j} P_{j|i} \log \frac{P_{j|i}}{Q_{j|i}})$ by a simple gradient descent (can be stuck in local minima).
- Parameters σ_i such that $H(P_i) = -\sum_{j=1}^n P_{j|i} \log P_{j|i} = \text{cst}$.

1. Dimension Reduction
   - Reconstruction error
   - Distance preservation

2. Feature Design
   - Renormalization
   - Basis and Dictionary Learning
   - Categorical Feature Encoding
   - Quantization and Binarization
   - Hashing
   - Pooling
   - Application Specific Features

Feature Types

- Quantitative feature :
  - univariate, multivariate, *functional*
  - continuous, discrete
- Categorical feature :
  - Binary, nominal, ordinal
- List, relationship...

- **Idea :** a good prediction algorithm should be invariant to change of scale in the variables.

Renormalization

- centering and standardization of a feature x,
- mapping of x to $[0, 1]$ if max and min are known

- Renormalization is *useless for purely linear methods*... but few methods are purely linear !
- **Idea :** Linear scale may not be the most natural one...

Transformation

- log-scale instead of linear scale,
- Box-Cox transform,
- sigmoid, maxout, rectifier...
- application dependent transformation.

- **Idea :** the behavior may not be *linear* in the feature.

Basis decomposition

- Replace a feature x by $\varphi(x) = (\varphi_1(x), \ldots, \varphi_B(x))$ where $(\varphi_b)_{b=1}^{B}$ is a linearly independent family of functions.
- In linear methods, use $\langle \alpha, \varphi(x) \rangle$ instead of αx
- Examples : polynomials, Fourier, wavelets...

- Non parametric estimation technique...

- **Extension :** the behavior may not be *linear* in some features.

Basis decomposition

- Replace some features $x = (x_1, \ldots, x_n)$ by $\varphi(x) = (\varphi_1(x), \ldots, \varphi_B(x))$ where $(\varphi_b)_{b=1}^{B}$ is a linearly independent family of functions.
- In linear methods, use $\langle \alpha, \varphi(x) \rangle$ instead of $\langle \alpha, x \rangle$
- Positive definite kernel associated to the scalar product $K(x, y) = \langle \varphi(x), \varphi(y) \rangle$ plays an important role.
- φ is often defined implicitly by a choice of K...

- **Kernel trick :** SVM but also (generalized) linear model as seen for example with *quadratic* logistic modeling.

Feature Design
Quantitative Feature Recombination

- **Idea :** some combinations of the features may be more interesting than any single one.

Decomposition idea

- Obtain $(x_1, \ldots, x_K)$ as a linear combination of some vectors $\varphi_1, \ldots, \varphi_{K'}$ of dimension K

$$\begin{pmatrix} x_1 \\ \vdots \\ x_K \end{pmatrix} \sim \sum_{k'=1}^{K'} x'_{k'} \varphi_{k'}$$

- Use $(x'_1, \ldots, x'_{K'})$ instead of $(x_1, \ldots, x_k)$

- **Variation :** use $(x'_1, \ldots, x'_{K''})$ with $K'' \leq K'$
- Most important part : choice of φ !

PCA

- $\varphi_1, \ldots, \varphi_K$ are the eigenvectors of the empirical covariance matrix of $(x_1, \ldots, x_K)$
- The φ_k are obtained by minimizing recursively :

$$\varphi_k = \operatorname*{argmin}_{\varphi} \frac{1}{n} \sum_{i=1}^{n} \min_{x' \in \mathbf{R}^k} \|(x_{1,n} \ldots, x_{K,n})^t - (\sum_{k'=1}^{k-1} x'_{k'} \varphi_{k'} + x'_k \varphi)\|^2$$

- The coefficents $x'_{k'}$ are obtained for a new feature by minimizing

$$\|(x_1 \ldots, x_K)^t - (\sum_{k'=1}^{K'} x'_{k'} \varphi_{k'})\|^2$$

- Change of basis ! Useless for purely linear method if all the coefficients are kept !

NMF (force positive weights !)

- Fix K' and find vectors $\varphi_1, \ldots, \varphi_{K'}$ such that the vectors of feature are well approximated by a linear sum with positive weights
- Non trivial minimization problem to find φ_k :

$$(\varphi_1, \ldots, \varphi_{K'}) = \operatorname*{argmin}_{(\varphi_1,\ldots,\varphi_{K'})} \frac{1}{n} \sum_{i=1}^{n} \min_{x' \in (\mathbf{R}^+)^{K'}} \|(x_{1,n} \ldots, x_{K,n})^t - (\sum_{k'=1}^{K'} x'_{k'} \varphi_{k'})\|^2$$

- The coefficients $x'_{k'}$ are obtained for a new feature by minimizing over $x' \in (\mathbf{R}^+)^{K'})$

$$\|(x_1 \ldots, x_K)^t - (\sum_{k'=1}^{K'} x'_{k'} \varphi_{k'})\|^2$$

- **Idea :** transform a categorical feature x in *quantitative* ones c (encoding)...

Encodings

- Binary encoding : $x \in \{C_0, C_1\}$
  - binary code : $c(x) = \mathbf{1}_{x=C_1}$
  - symmetrized version : $c(x) = 2\mathbf{1}_{x=C_1} - 1$
- Nominal variable : $x \in \{C_1, \ldots, C_V\}$
  - binary code : $c(x) = (\mathbf{1}_{x=C_2}, \ldots, \mathbf{1}_{x=C_V})$
  - $V - 1$ quantitative variables.
- Ordinal variables : $x \in \{C_1, \ldots, C_V\}$
  - binary code : $c(x) = (\mathbf{1}_{x \geq C_2}, \ldots, \mathbf{1}_{x \geq C_V})$
  - $V - 1$ quantitative variables.
  - Feature selection makes more sense with those feature.

General coding scheme

- Matrix representation :

$$c(x)^t = M \begin{pmatrix} \mathbf{1}_{x=C_0} \\ \vdots \\ \mathbf{1}_{x=C_V} \end{pmatrix}$$

with $M \in \mathcal{M}_{V-1,V}$ chosen such that its column are linearly independent.

- Examples for $V = 2$:

$$M = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad M = \begin{pmatrix} -1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix}$$

$$M = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \qquad M = \begin{pmatrix} -1/3 & 2/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{pmatrix}, \ldots$$

- Choice of the coding matters for feature selection

- **Idea :** Capture interaction between features by encoding them jointly.

Full interaction between $x = (x_1, \dots, x_K)$

- Matrix encoding :

$$c(x)^t = M \begin{pmatrix} \mathbf{1}_{x=(C_{1,1},\dots,C_{1,K-1},C_{1,K})} \\ \mathbf{1}_{x=(C_{1,1},\dots,C_{1,K-1},C_{V_K,K})} \\ \mathbf{1}_{x=(C_{1,1},\dots,C_{2,K-1},C_{1,K})} \\ \vdots \\ \mathbf{1}_{x=(C_{V_1,1},\dots,C_{V_{K-1},K-1},C_{V_K,K})} \end{pmatrix}$$

with $M \in \mathcal{M}_{\prod_{k=1}^K V_k - 1, \prod_{k=1}^K V_k}$ such that the lines are linearly independent...

- **Example** of code for $x = (x_1, x_2)$ with respectively 2 and 3 categories :

$$c(x)^t = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{1}_{x=(1,1)} \\ \mathbf{1}_{x=(1,2)} \\ \mathbf{1}_{x=(1,3)} \\ \mathbf{1}_{x=(2,1)} \\ \mathbf{1}_{x=(2,2)} \\ \mathbf{1}_{x=(2,3)} \end{pmatrix}$$

- Interaction of order K leads to a very large number of categories !

- Modified coding scheme to impose a **hierarchical structure** :

$$c(x)^t = M \begin{pmatrix} \mathbf{1}_{x_1=c_{1,1}} \\ \vdots \\ \mathbf{1}_{x_K=c_{V_K,K}} \\ \mathbf{1}_{(x_1,x_2)=(c_{1,1},c_{1,2})} \\ \vdots \\ \mathbf{1}_{(x_{K-1},x_K)=(c_{V_{K-1},K-1},c_{V_K,K})} \\ \vdots \\ \\ \vdots \\ \mathbf{1}_{x=(c_{1,1},\dots,c_{1,K-1},c_{1,K})} \\ \vdots \\ \mathbf{1}_{x=(c_{V_1,1},\dots,c_{V_{K-1},K-1},c_{V_K,K})} \end{pmatrix}$$

with $M \in \mathcal{M}_{\prod V_k - 1, \sum_{k=1}^{K} \sum_{i_1 < \dots < i_{k'} < \dots < i_k} \prod V_{i_{k'}}}$ such that :

  - its lines are linearly independent
  - the first $(\sum_{i_1 < \dots < i_{k'} < \dots < i'_K} \prod V_{i_{k'}}) - 1$ below are equal to zeros after the column $\sum_{k=1}^{K} \sum_{i_1 < \dots < i_k} \prod V_{i_k}$

Feature Design

Categorical Feature Interaction

- **Example** with $x = (x_1, x_2)$ with respectively 2 and 3 categories

$$c(x)^t = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{1}_{x_1=1} \\ \mathbf{1}_{x_1=2} \\ \mathbf{1}_{x_2=1} \\ \mathbf{1}_{x_2=2} \\ \mathbf{1}_{x_3=3} \\ \mathbf{1}_{x=(1,1)} \\ \mathbf{1}_{x=(1,2)} \\ \mathbf{1}_{x=(1,3)} \\ \mathbf{1}_{x=(2,1)} \\ \mathbf{1}_{x=(2,2)} \\ \mathbf{1}_{x=(2,3)} \end{pmatrix}$$

- Restriction possible to a given order of interaction by using only the first lines !
- Coding variant for ordered categories.
- **Extension** to interaction between a quantitative feature x_1 and a categorical feature x_2 through the mapping

$$(x_1, x_2) \to (x_1, x_1 c(x_2))$$

- **Idea :** Going from quantitative feature to binary features...
- Used in practice mainly for computing reason.

Quantization/Binarization strategy

- Construct a finite size quantifier for the feature.
- Code the feature by a binary code of its quantized version.

- Construction of a quantifier :
  - Binning : use of a (regular) histogram with V
  - V-means : use the centers as *keywords* and assign a point to its nearest keyword.
  - Use $V-1$ quantiles
  - Vector coding...
- Extreme case : $V=2$ binarization !

- **Idea :** Reduce the number of values of a nominal feature with categories in a large set $\mathcal{D}$?

Hashing

- Construction of a *hashing* function $H : \mathcal{D} \to \{1, \ldots, V\}$ and use the hashed value instead of the original one.
- The hashing function should be *as injective as possible...* in a probabilistic sense..

- Design of such hashing function is a real art !
- One can sometimes find hashing function such that $d(H(C_k), H(C_{k'})) \sim d'(C_k, C_{k'})$...

- **Idea :** Group features in a non linear way to reduce the number of features.

Pooling

- **Quantitative features :** replace a subset or a list by
  - its min/max,
  - its range,
  - its average (linear...),
  - ...
- **Nominal features :** replace a subset or a list by
  - its histogram
  - its most frequent values
  - ...

- Discretization/Binarization can be use before and after...

- Use of **application field specific** features :
  - Lots of know-how by experts of the field
  - (Almost) no price of using them if good feature selection

Two examples

- Text and bag of words
- Image and SIFT (Scale Invariant Feature Transform)

- Deep Neural Networks seem not to require this part...

- How to transform a **text** into a vector of **categorical features**?

Bag of Words strategy

- Make a *list* of words,
- Compute a *weight* for each words.

List buiding

- Make a list of all used words with their number of occurrence
- Gather the words in the list having the same stem (stemming)
- Hash the stem using a word specific hashing function (MurmurHash with 32bits for instance)
- Compute the histogram $h_w(d)$

Weight computation

- Compute the histogram $h_w(d)$
- Apply the the tf-idf transform to the histogram

$$\mathrm{tf} - \mathrm{idf}_w(d) = \mathrm{idf}_w \times \mathrm{tf}_w(d)$$

with idf a corpus dependent weight

$$\mathrm{idf}_w = \log \frac{n}{\sum_{i=1}^{n} \mathbf{1}_{h_w(d_i) \neq 0}}$$

and $\mathrm{tf}_w(d)$ the frequency within the document d

$$\mathrm{tf}_w(d) = \frac{h_w(d)}{\sum_w h_w(d)}.$$

- Most classical text preprocessing !

- How to transform an **image** into a vector of **features** ?

SIFT Strategy

- Compute a local descriptor based on local gradient.
- Agregate those measurements by histograms.

SIFT Local Descriptor

- Compute a local scale and a principal orientation
- Compute the gradient at that scale, its norm and its angle with the principal direction
- Quantize this angle with 16 different values (binning)
- On each subwindow of 4×4 subwindow grid oriented with the principal and of size the local scale, compute the sum of the gradient norm for each angle bin renormalized by the number of points in the subwindow.
- Use the $4 \times 4 \times 16$ values as the local descriptor.

SIFT based representation

- Compute the SIFT descriptor at each point of a regular grid
- Assign each SIFT descriptor to the closest *keyword* obtained by K-means on the whole corpus (typically with $K \sim 2000$)
- Compute a normalized histogram of the counts of those keywords
- Use this 2000 values as the image descriptor

- Used to be state-of-the-art, now lags behing DNN...