Transductive Learning and Computer Vision

Jean-Yves Audibert[1]

joint works with

Olivier Duchenne
Patrick Etyngier
Matthias Hein
Renaud Keriven
Ulrike von Luxburg
Jean Ponce
Hichem Sahbi

[1] Willow project & Certis - Ecole des Ponts - Paris Est

Learning with Data-dependent Concept Spaces, 2008

Outline

Inductive, semi-supervised and transductive learning

Inductive learning

- We observe $(X_1, Y_1), \ldots, (X_n, Y_n)$
- Question: For any new input x, what is the associated y?

Semi-supervised learning

- We observe $(X_1, Y_1), \ldots, (X_n, Y_n)$ and $X_{n+1}, \ldots, X_{n+t}$
- Question: For any new input x, what is the associated y?

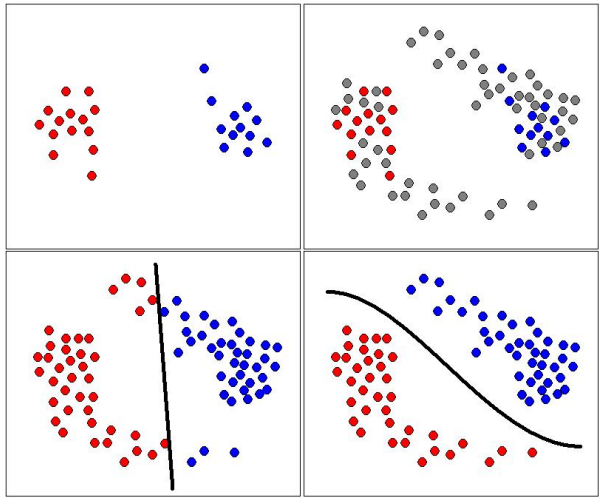
Transductive learning

- We observe $(X_1, Y_1), \ldots, (X_n, Y_n)$ and $X_{n+1}, \ldots, X_{n+t}$
- Question: what are the outputs $Y_{n+1}, \ldots, Y_{n+t}$?

The key semi-supervised and transductive learning assumption

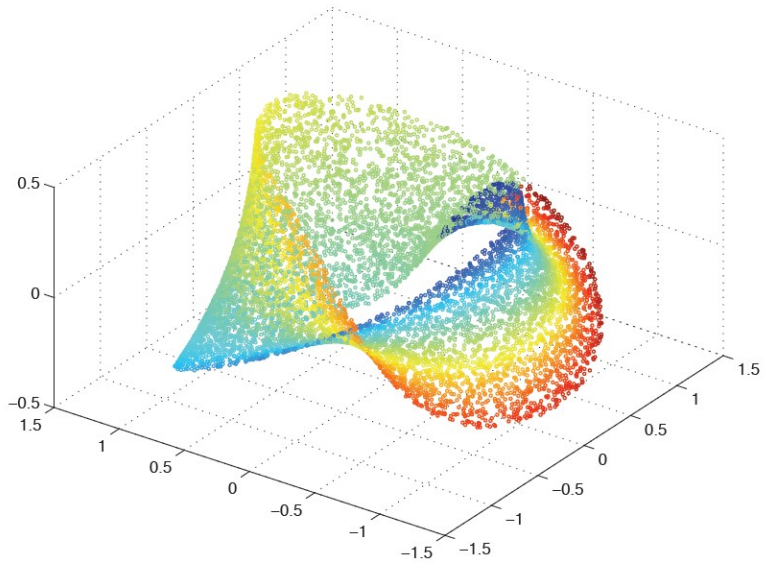
Key assumption

The decision boundary appears in a low density region of the input space.



Input data distribution

- $X_1, X_2, \ldots$ drawn i.i.d. from $P(dx) = p(x)dV_M(x)$
- V_M : natural volume element



- $M \subset \mathbb{R}^Q$: unknown submanifold of dimension $q \leq Q$

Laplace operator

- If $f : \mathbb{R} \to \mathbb{R}$ twice differentiable:

$$\Delta f = f''$$

- If $f : \mathbb{R}^Q \to \mathbb{R}$ twice differentiable:

$$\Delta f = \sum_{i=1}^{Q} \frac{\partial^2 f}{\partial x_i^2}$$

- For M a submanifold of $\mathbb{R}^Q$ and $f : M \to \mathbb{R}$ twice differentiable:

$$\Delta f = \operatorname{div}(\operatorname{grad} f)$$

The s-weighted Laplace-Beltrami operator

- M a submanifold of $\mathbb{R}^Q$
- p a density on M
- $s \in \mathbb{R}$
- $f : M \to \mathbb{R}$ twice differentiable:

$$\Delta_s f := \frac{1}{p^s} \operatorname{div}(p^s \operatorname{grad} f)$$

- Main properties:
  - $\Delta_0 = \Delta$
  - $\forall f, g \quad \int_M f(\Delta_s g)\, p^s dV_M = -\int_M \langle \nabla f, \nabla g \rangle\, p^s dV_M$

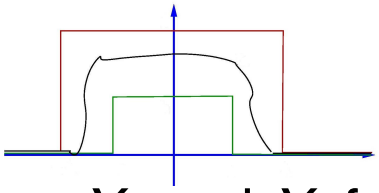
Problem

Given $f : M \to \mathbb{R}$, how to approximate $(\Delta_s f)(x)$ by only using the values of f at $x, X_1, \ldots, X_n$?

Motivation

- Clustering (Spielman & Teng, 1996, ...)
- Dimensionality reduction (Belkin & Niyogi, 2003, ...)
- Transductive learning (Belkin & Niyogi, 2004, ...)

Neighbourhood graph



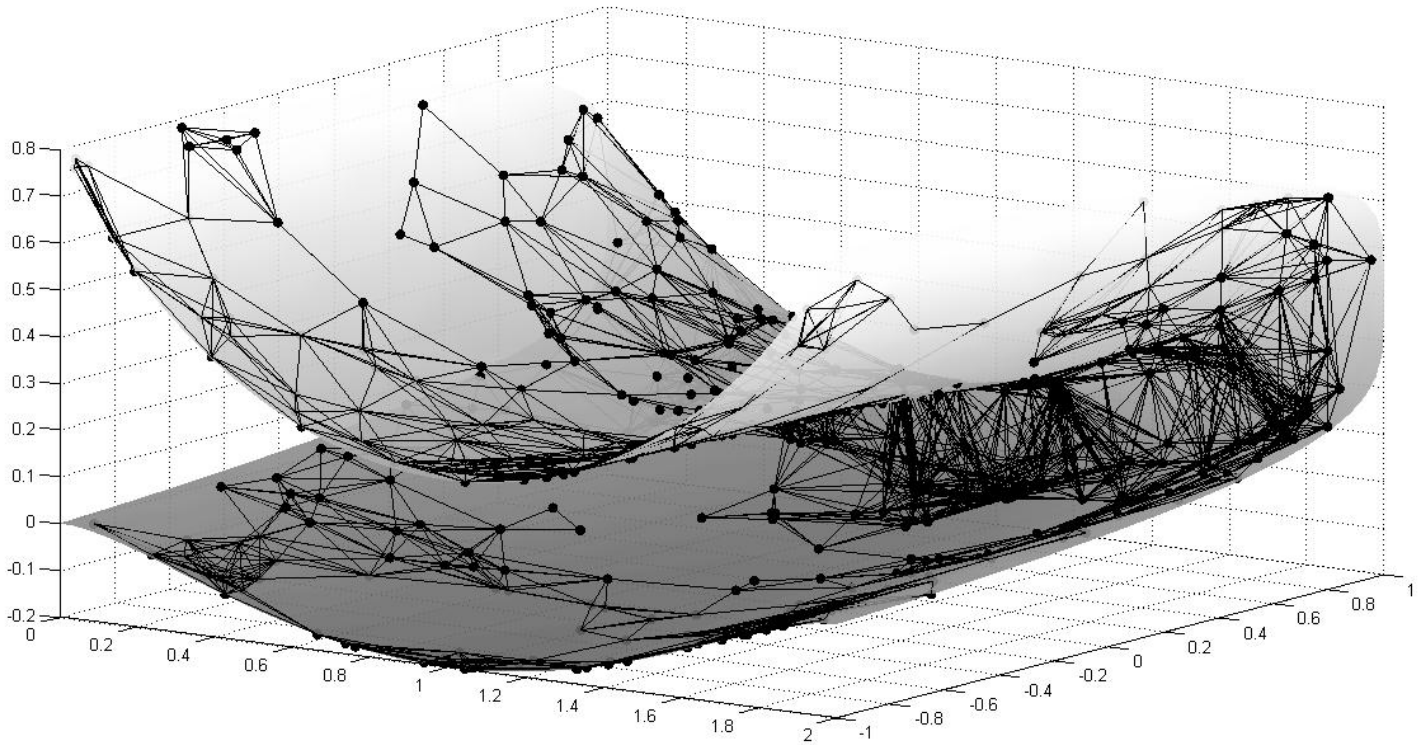
- $\tilde{k} : \mathbb{R}_+ \to \mathbb{R}_+$ such that $k(u) = 0$ for $u \geq 1$
- $\tilde{k}(\|X_i - X_j\|^2/h^2)$: similarity measure between X_i and X_j for the bandwith parameter $h > 0$
- $\tilde{d}(X_i) = \sum_{j=1}^{n} \tilde{k}(\|X_i - X_j\|^2/h^2)$
- $k(X_i, X_j) = \frac{\tilde{k}(\|X_i - X_j\|^2/h^2)}{[\tilde{d}(X_i)\tilde{d}(X_j)]^{\lambda}}$. λ = reweighting parameter

Definition

The neighbourhood graph:

- $V = \{X_1, \ldots, X_n\}$
- $E = \{(X_i, X_j) : k(X_i, X_j) > 0\}$
- $w(X_i, X_j) = k(X_i, X_j)$

Neighbourhood graph



Extended neighbourhood graph

- Set of vertices: $V = \{x, X_1, \ldots, X_n\}$
- Degree function: $d(x) = \sum_{j=1}^{n} k(x, X_j)$
- Averaging operator: for any $f : M \to \mathbb{R}$,

$$(Af)(x) = \sum_{j=1}^{n} k(x, X_j) f(X_j).$$

Definition of the "random walk" graph Laplacian

- Random walk: $(\Delta^{(\mathrm{rw})} f)(x) = \frac{1}{h^2}\left(f - \frac{1}{d} A f\right)(x)$
- Similar to:

$$\begin{aligned} f''(x) &\approx \frac{-2f(x)+f(x-h)+f(x+h)}{h^2} \\ &\propto \frac{1}{h^2}\left(f(x) - \frac{f(x-h)+f(x+h)}{2}\right) \end{aligned}$$

- Let $X_0 = x$ and

$$T = \left(\frac{k(X_i, X_j)}{\sum_{\ell=0}^{n} k(X_i, X_\ell)}\right)_{0 \le i,j \le n}$$

- $T_{ij} \geq 0$ and $\sum_{j=0}^{n} T_{ij} = 1 \Rightarrow T$ is the transition matrix

$$(\Delta^{(\mathrm{rw})} f)(x) = \frac{1}{h^2}\left(f(x) - \mathbb{E}_{W_1 | W_0 = x} f(W_1)\right).$$

Definitions

$$\text{random walk} \quad (\Delta^{(\mathrm{rw})} f)(x) = \frac{1}{h^2}\left(f - \frac{1}{d}Af\right)(x)$$

$$\text{unnormalized} \quad (\Delta^{(\mathrm{u})} f)(x) = \frac{(nh^q)^{2\lambda-1}}{h^2}\Big(df - Af\Big)(x)$$

$$\text{normalized} \quad (\Delta^{(\mathrm{n})} f)(x) = \frac{1}{h^2\sqrt{d(x)}}\left(d\frac{f}{\sqrt{d}} - A\Big(\frac{f}{\sqrt{d}}\Big)\right)(x)$$

$$= \frac{1}{h^2}\left(f - \frac{1}{\sqrt{d}}A\Big(\frac{f}{\sqrt{d}}\Big)\right)(x)$$

Convergence properties

Graph Laplacian convergence (Hein, Audibert & von Luxburg, 2005)

Let $s = 2(1-\lambda)$. Under reasonable conditions on the submanifold M, the kernel $\tilde{k}$ and the density p:

- if $h \to 0$ and $nh^{q+2}/\log n \to \infty$,

$$\text{random walk:} \quad \lim_{n\to\infty} (\Delta^{(\mathrm{rw})} f)(x) \propto -(\Delta_s f)(x) \quad \text{a.s.}$$

$$\text{unnormalized:} \quad \lim_{n\to\infty} (\Delta^{(\mathrm{u})} f)(x) \propto -p(x)^{1-2\lambda} (\Delta_s f)(x) \quad \text{a.s.}$$

- if $h \to 0$ and $nh^{q+4}/\log n \to \infty$,

$$\text{normalized:} \quad \lim_{n\to\infty} (\Delta^{(\mathrm{n})} f)(x) \propto -p(x)^{\frac{1}{2}-\lambda} \Delta_s \left(\frac{f}{p^{\frac{1}{2}-\lambda}} \right)(x) \quad \text{a.s.}$$

Convergence properties: discussion

- The graph Laplacian converges to the true Laplacian (up to some terms)!
- All limits agree for $\lambda = 1/2$
- All limits agree for uniform density
- In other cases, the limits are different!
- The data-dependent modification of the edge weights allows to control the influence of the density
- Dependence on q of $(\Delta^{(\mathrm{u})} f)(x) = \frac{(nh^q)^{2\lambda-1}}{h^2}\Big(df - Af\Big)(x)$

A way of using the key assumption

- Recalling the key assumption: The input density p at the decision boundary is small
- Incorporation of the prior knowledge:

$$\min_f \; c \sum_{1 \le i \le n} [Y_i - f(X_i)]^2 + \int_M \|\nabla f\|^2 p^s dV_M,$$

$$\Leftrightarrow \min_f \; c \sum_{1 \le i \le n} [Y_i - f(X_i)]^2 - \int_M f \times (\Delta_s f)\, p^s dV_M,$$

$$\underset{\approx}{\Leftrightarrow} \min_f \; c \sum_{1 \le i \le n} [Y_i - f(X_i)]^2 - \frac{1}{n+t} \sum_{i=1}^{n+t} f(X_i) \Delta_s f(X_i) p^{s-1}(X_i),$$

A way of using the key assumption

- Recalling the key assumption: The input density p at the decision boundary is small
- Incorporation of the prior knowledge:

$$\min_f \; c \sum_{1 \le i \le n} [Y_i - f(X_i)]^2 + \int_M \|\nabla f\|^2 p^s dV_M,$$

$$\Leftrightarrow \min_f \; c \sum_{1 \le i \le n} [Y_i - f(X_i)]^2 - \int_M f \times (\Delta_s f)\, p^s dV_M,$$

$$\underset{\approx}{\Leftrightarrow} \; \min_f \; c \sum_{1 \le i \le n} [Y_i - f(X_i)]^2 - \frac{1}{n+t} \sum_{i=1}^{n+t} f(X_i) \Delta_s f(X_i) p^{s-1}(X_i),$$

A way of using the key assumption

- Recalling the key assumption: The input density p at the decision boundary is small
- Incorporation of the prior knowledge:

$$\min_f \; c \sum_{1 \le i \le n} [Y_i - f(X_i)]^2 + \int_M \|\nabla f\|^2 p^s dV_M,$$

$$\Leftrightarrow \min_f \; c \sum_{1 \le i \le n} [Y_i - f(X_i)]^2 - \int_M f \times (\Delta_s f)\, p^s dV_M,$$

$$\underset{\approx}{\Leftrightarrow} \min_f \; c \sum_{1 \le i \le n} [Y_i - f(X_i)]^2 - \frac{1}{n+t} \sum_{i=1}^{n+t} f(X_i) \Delta_s f(X_i) p^{s-1}(X_i),$$

Approximating the graph Laplacian

- Now we have seen that $\Delta^{(\mathrm{u})} f \rightsquigarrow -p^{s-1}\,\Delta_s f$. Therefore:

$$\cdots \underset{\approx}{\Leftrightarrow} \min_f \; c \sum_{1 \le i \le n} [Y_i - f(X_i)]^2 - \tfrac{1}{n+t} \textstyle\sum_{i=1}^{n+t} f(X_i) \Delta^{(\mathrm{u})} f(X_i),$$

- Besides we have $\left((\Delta^{(\mathrm{u})} f)(X_i)\right)_{1 \le i \le n+t} \propto (D - W)F$, where
  - D is the diagonal matrix with $D_{ii} = d(X_i)$
  - W is the weight matrix with $W_{ij} = k(X_i, X_j)$
  - $F = (f(X_i))_{1 \le i \le n+t}$ is the predicted output vector

A linear system to solve

- Let $Y = (Y_1, \ldots, Y_n, 0, \ldots, 0)^T \in \mathbb{R}^{n+t}$
- Let $C = Diag(c, \ldots, c, 0, \ldots, 0) \in \mathbb{R}^{(n+t)\times(n+t)}$
- The predicted output associated to the test points are the last m elements of the vector F solving

$$\min_{F \in \mathbb{R}^{n+t}} \; (F - Y)^T C (F - Y) + F^T (D - W) F,$$

- By differentiation, the solutions satisfy

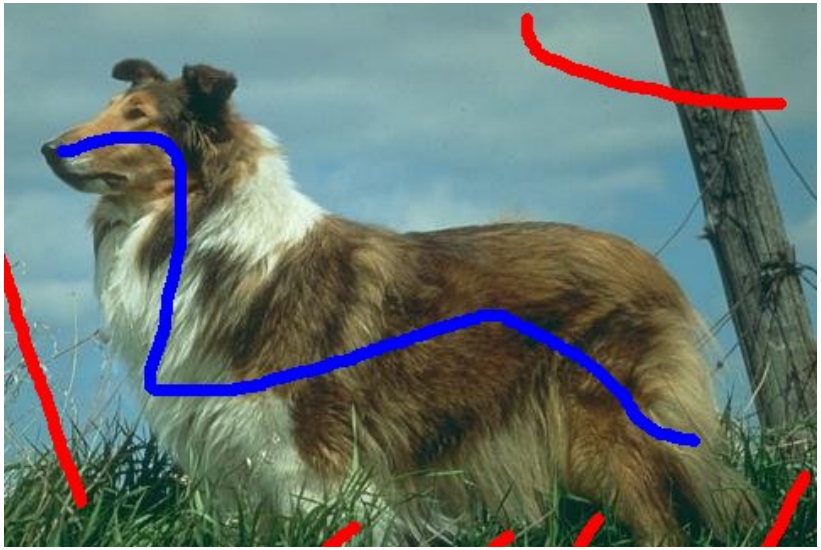
$$(D - W + C)F = CY$$

- For the unlabeled input X_i, output $\mathrm{sgn}(F_i)$
- 3 parameters: h, c, and s

Image segmentation

- Partitioning an image into “meaningful” regions
- A key task of computer vision (medical imaging, photo/painting softwares,...)
- An ill-posed problem $\Rightarrow$ utilize user-supplied seeds

Image segmentation



Algorithm (1/2)

(Duchenne, Audibert, Keriven, Ponce, Segonne, 2008)

Parameters

- $s \geq 0$
- $\sigma_g > 0$: scale of geometric neighbourhoods
- $\sigma_c > 0$: scale of chromatic neighbourhoods
- $m \in \mathbb{N}$: size of the local patch

Let

- $C(i)$ = the RGB levels of a square patch of size $2m+1$ around the pixel i.
- $z(i)$ = the geometric position (row+column) of the pixel i

$$\tilde{k}(i,j) = e^{-\frac{\|z(i)-z(j)\|^2}{2\sigma_g^2} - \frac{\|C(i)-C(j)\|^2}{2\sigma_c^2}}.$$

Algorithm (2/2)

- Compute the matrix D and W:

$$W_{i,j} = \frac{\tilde{k}(i,j)}{[\tilde{d}(i)\tilde{d}(j)]^{1-s/2}} \qquad D_{i,i} = \sum_j W_{i,j}$$

- For the training pixel i, put $Y_i = -1$ or $+1$ depending on which zone the pixel i belongs to
- Solve the large sparse linear system associated to

$$\min_{\substack{F \in \mathbb{R}^{n+t} \\ \forall i \in [[1,n]] F_i = Y_i}} F^T(D-W)F$$

- Output for the pixel j the label $\mathrm{sgn}(F_j)$

Link to previous approaches

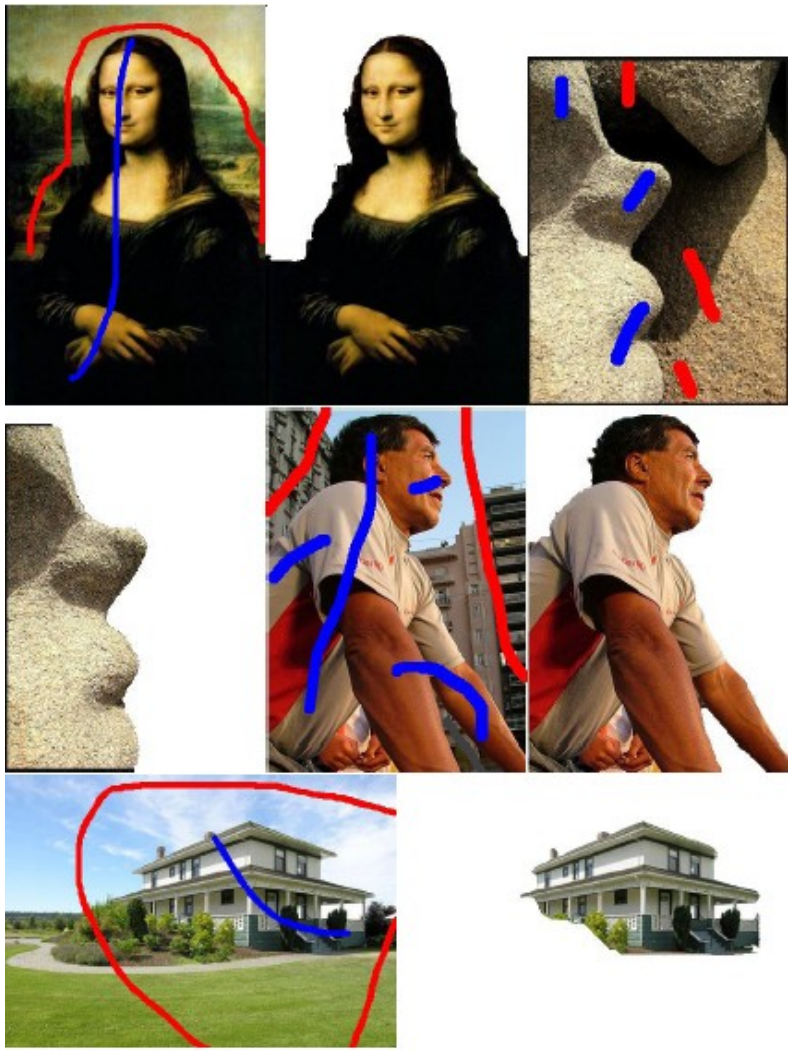
- Our approach: $\min_{\substack{F \in \mathbb{R}^{n+t} \\ \forall i \in [[1,n]] F_i = Y_i}} F^T \Delta^{(u)} F$ with $\Delta^{(u)} = D - W$
- Graph cut: $\min_{\substack{F \in \{-1;+1\}^{n+t} \\ \forall i \in [[1,n]] F_i = Y_i}} F^T \Delta^{(u)} F$

  For $s = 2$, Boykov et al. (2001,2006), Blake et al. (2006)
- Normalized cut: $s = 1 \rightarrow \Delta^{(u)}$ is the matrix of the eigenvalue problem used in (Shi & Malik, 2000)
- Guan and Qiu (2006): $\min_{\substack{F \in \mathbb{R}^{n+t} \\ \forall i \in [[1,n]] F_i = Y_i}} F^T \Delta^{(rw)^2} F.$

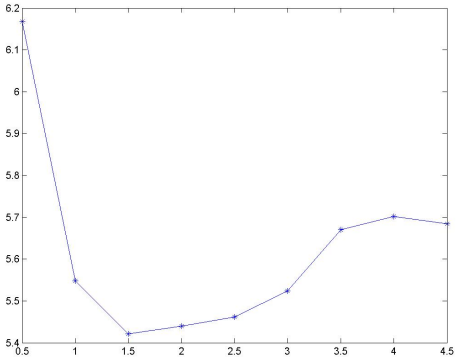
$$\int_M \|\nabla f\|^2 p^s dV_M \neq \int_M |\Delta f|^2 p^s dV_M$$

- Grady et al. (2004,2006): $s = 2$ (motivated by graph-theoretical electrical potential)

Some experimental results

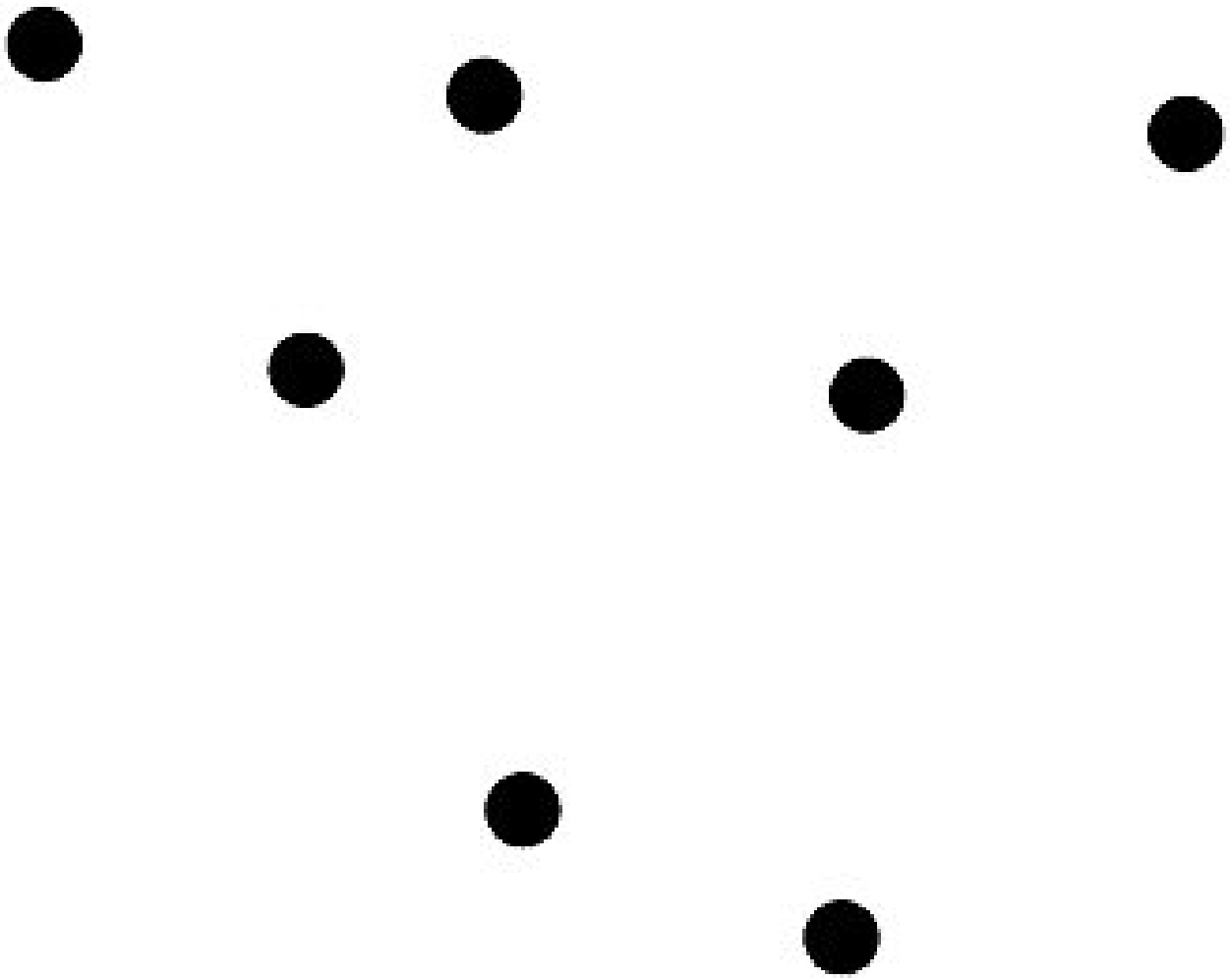


Some experimental results

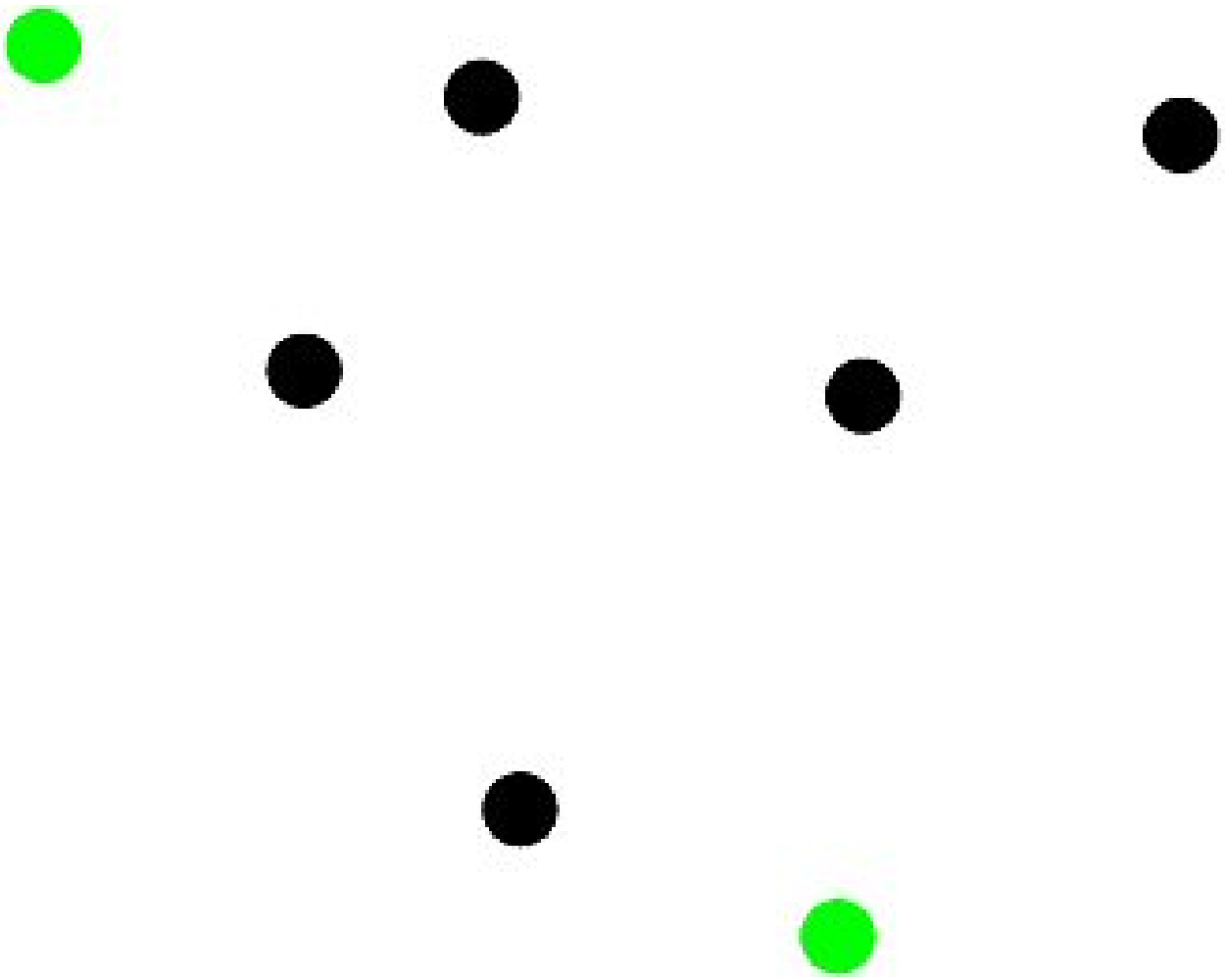


- Quantatively:
  5.4% of the pixel of the grey band are misclassified in average

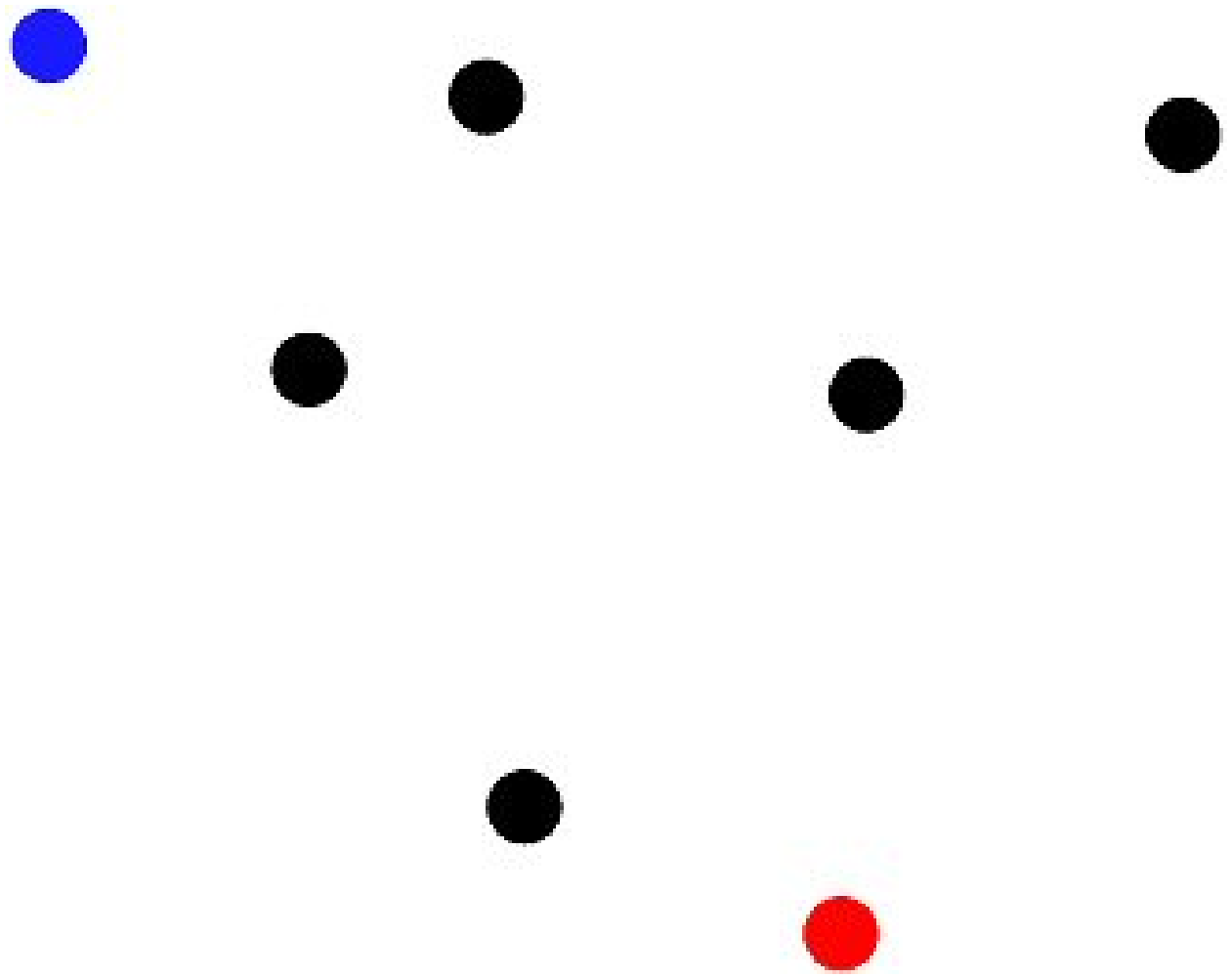
Principle



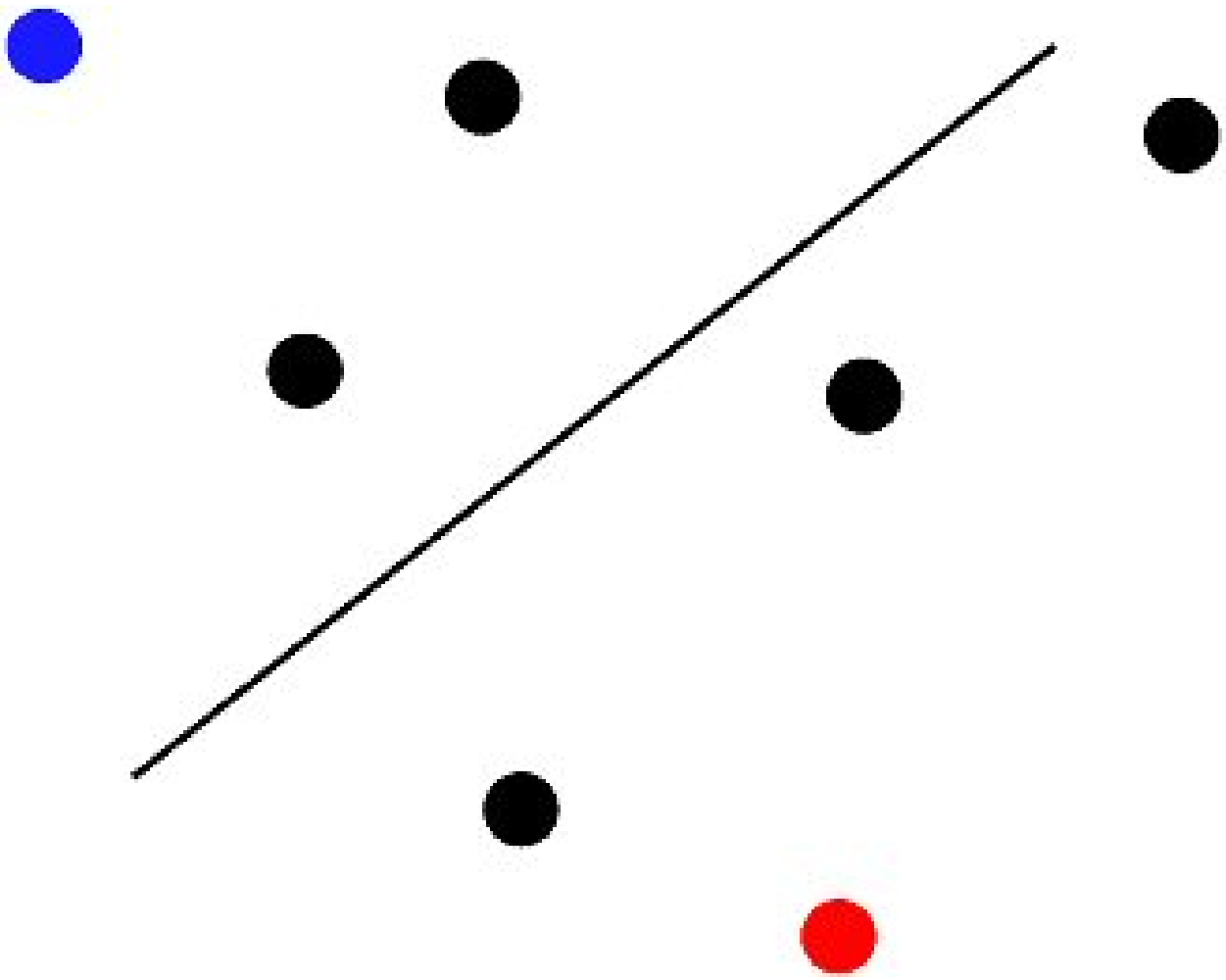
Principle



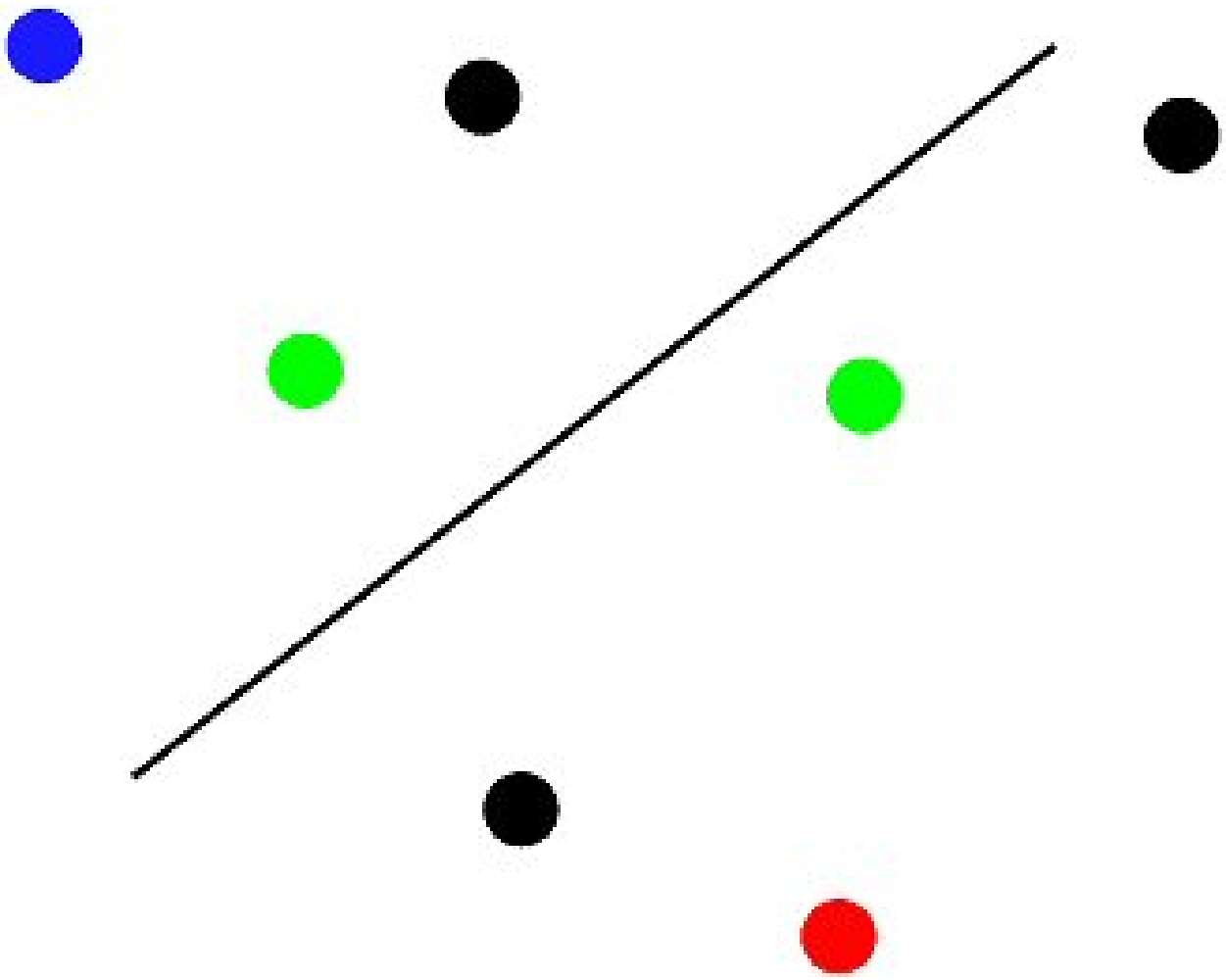
Principle



Principle



Principle



Principle & Evaluation

- Principle
  (1) Display initial images
  (2) Ask the user for the interesting/non-interesting ones
  (3) Display new images and goto (2)
- Evaluation: recall vs number of displayed images

Random walks

- W weight matrix: $W_{i,j} = k(X_i, X_j)$, $i, j \in [[1, n]]$
- D diagonal matrix: $D_{i,i} = \sum_j W_{i,j}$
- $P = D^{-1} W$ is a transition matrix
- $(\lambda_1, \psi_1), \ldots, (\lambda_n, \psi_n)$ eigenvalues and eigenvectors of P
- $X_i \to \phi(X_i) = (\psi_{1,i}, \ldots, \psi_{n,i})$
- Diffusion distance:

$$D_{\mathcal{M}}(X_i, X_j) = \|P(\cdot|i) - P(\cdot|j)\|^2_{\ldots} = \sum_{\ell=1}^{n} \lambda_\ell^2 \left[\phi_\ell(X_i) - \phi_\ell(X_j)\right]^2$$

$$\approx \sum_{\ell=1}^{L} \lambda_\ell^2 \left[\phi_\ell(X_i) - \phi_\ell(X_j)\right]^2$$

Random walks

- $P_k = P^k$ is a (k-step) transition matrix

$$P_k(j|i) = \sum_{\ell_1,\ldots,\ell_{k-1}} P(\ell_1|i)P(\ell_2|\ell_1)\cdots P(j|\ell_{k-1})$$

- k-step diffusion distance:

$$D_{k,\mathcal{M}}(X_i, X_j) = \|P_k(\cdot|i) - P_k(\cdot|j)\|^2_{\ldots} = \sum_{\ell=1}^{n} \lambda_\ell^{2k}\left[\phi_\ell(X_i) - \phi_\ell(X_j)\right]^2$$

$$\approx \sum_{\ell=1}^{L} \lambda_\ell^{2k}\left[\phi_\ell(X_i) - \phi_\ell(X_j)\right]^2$$

Exploitation-exploration

- half exploitation and half exploration for the new display
- half of the images are drawn according to

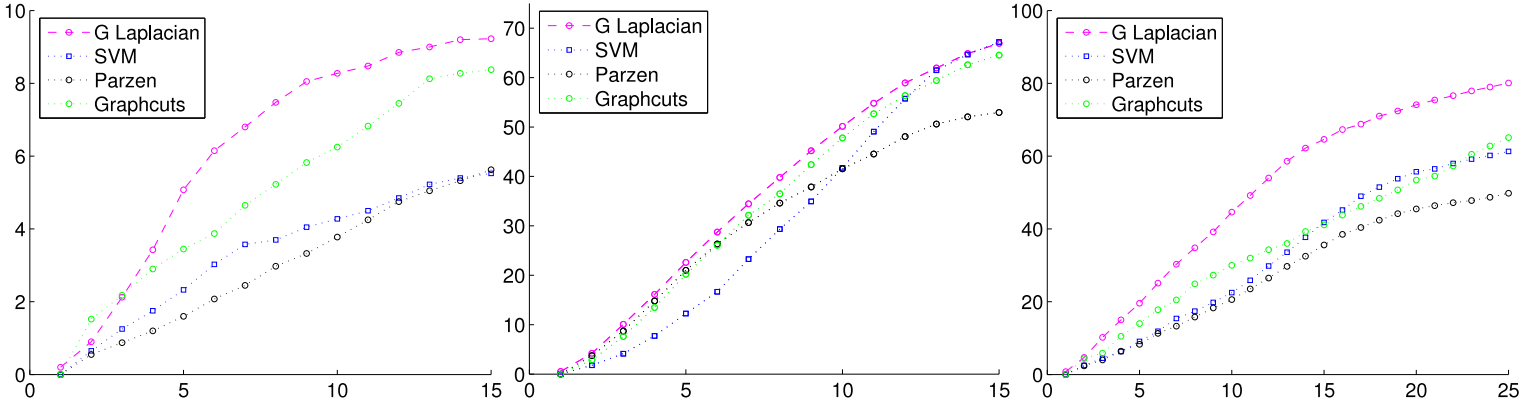
$$p(X_j) \propto \min_{i \text{ positively labeled}} D_{k,\mathcal{M}}(X_i, X_j)$$

- half of the images are drawn according to

$$p(X_j) \propto \max_{i \text{ positively labeled}} D_{k,\mathcal{M}}(X_i, X_j)$$

Experimental results

- Olivetti: 40 persons, 10 faces per person. Histogram equalization, KPCA -> 20 components kept
- Swedish: 15 categories of leaf silhouettes, 75 leafs per category. KPCA -> 14 components kept
- Corel: 90 categories, 100 images per category. 3D RGB histogram of 125 dimensions.



Conclusion

- Neighbourhood graph → graph Laplacian → transductive learning → image segmentation
- Neighbourhood graph → diffusion distance → interactive image search
- Take care when choosing a graph Laplacian matrix
  - several possibilities
  - normalization of the similarities
- Convergence results → new interpretation existing algorithms