Indépendance Conditionnelle et Prédiction

François Fleuret
École Polytechnique Fédérale de Lausanne, CVLab

Habilitation à Diriger des Recherches
Université Paris-13, 12 décembre 2006

Détection rapide de visages

collaboration avec D. Geman

Détection rapide de visages
Introduction

Localiser précisément tous les visages visibles dans une image noir et blanc.

Détection rapide de visages
Notations

La tâche principale consiste à prédire si une sous-image I contient un visage. Soient

- I à valeurs dans $[0,1]^{64\times 64}$ la sous-image,
- C à valeurs dans $\{0,1\}$ la présence du visage,
- $f : [0,1]^{64\times 64} \rightarrow \{0,1\}$ le prédicteur.

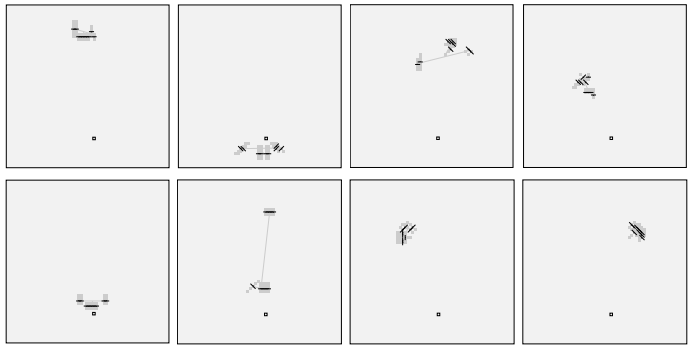
On cherche à construire un f qui minimise $P(f = 1|C = 0)$ sous $P(f = 0|C = 1) \leq \epsilon$.

La loi *a priori* est très déséquilibrée $P(C = 1) \ll 1$.

Détection rapide de visages
Pose non contrainte (1)

Notons

- $\xi_1, \ldots, \xi_B$ des détecteurs de bords à réponses binaires,
- $\forall \alpha \subset \{1, \ldots, B\},\ A(\alpha) = \prod_{i \in \alpha} \xi_i$
- $\mathcal{A}_1 = \{\{b\}, b \in \{1, \ldots, B\}\}$
- $\mathcal{A}_{2k} = \{\alpha \cup \alpha', (\alpha, \alpha') \in \mathcal{A}_k^2, \rho\,(A(\alpha), A(\alpha') \mid C = 1)) \geq \zeta\}$



Détection rapide de visages
Pose non contrainte (2)

Nous avons

$$\forall \alpha \in \mathcal{A}_k, P(A(\alpha)|C = 1) \geq \min_b P(\xi_b = 1|C = 1)\, \zeta^{\log_2 k}$$

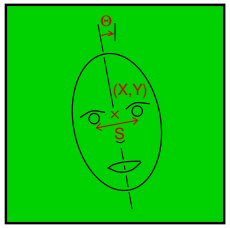
ce qui motive des classifieurs de la forme

$$f = 1_{\sum_j A(\alpha_j) \geq T}$$

avec $\forall j, \alpha_j \in \mathcal{A}_k$.

Détection rapide de visages
Pose contrainte (1)

Soit $\Gamma = (X, Y, \Theta, S)$ la pose du visage.



Sous hypothèse que les détecteurs de bords sont indépendants conditionnellement à $\Gamma = \gamma$, nous considérons

$$f_\gamma = 1_{\{\sum_{a \in A} \xi_k \geq T\}}$$

L'ensemble A étant choisi empiriquement pour maximiser T sous $\hat{P}(f = 0|C = 1) \simeq 0$

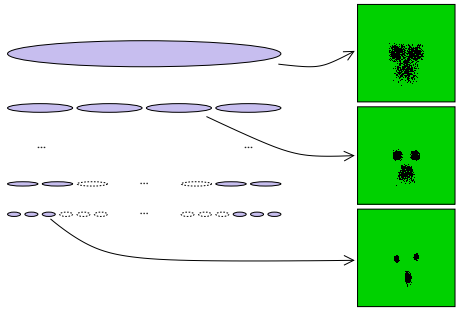
Détection rapide de visages
Pose contrainte (2)

Soit R_0^0 les poses admissibles pour les visages recherchés.

$$R_0^0 = \left\{(X, Y) \in [28; 36]^2, \Theta \in [-20; 20], S \in [10; 20]\right\}$$

Nous en considérons une partition récursive

$\forall a, \cup_b R_b^a = R_0^0$
$\forall a, \forall b \neq b', R_b^a \cap R_{b'}^a = \emptyset$
$\forall a > 0, \forall b, \exists b', R_b^a \subset R_{b'}^{a-1}$



Détection rapide de visages
Évaluation optimale

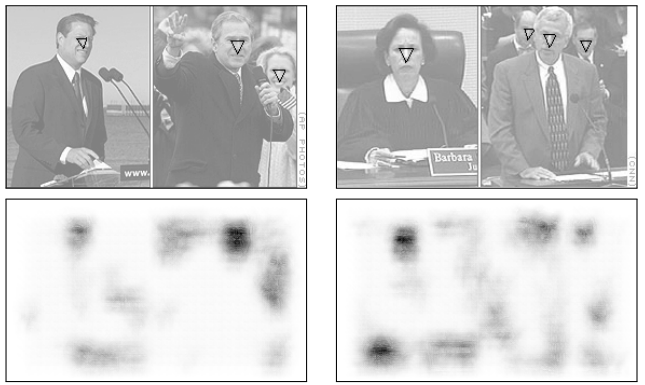
Nous associons un classifieur f_b^a à chacune de ces cellules et définissons une détection par

$$\exists b_0, \ldots, b_D, \forall d, f_{b_d}^d = 1$$

Si le coût algorithmique $C(f_b^a)$ est une fonction convexe de $P(f_b^a = 0|C = 0)$ alors n'évaluer un classifieur que si tous ses prédécesseurs ont répondu positivement est une stratégie optimale en moyenne.

Détection rapide de visages
Résultats (1)

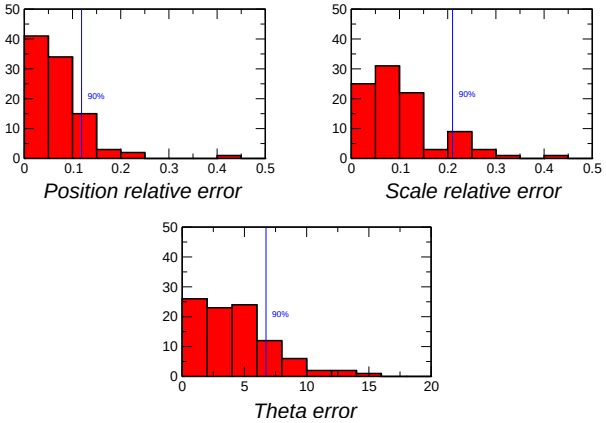
L'intensité du calcul est plus importante sur les parties ambiguës de l'image.



Le temps de calcul est de $\simeq 0.1$s pour une image 640×480 sur un processeur à 2.5Ghz.

Détection rapide de visages
Résultats (2)

Avec 400 images de visages pour l'apprentissage, aucune image de "fond", pour $\hat{P}(f = 0|C = 1) = 0.09$, le nombre de fausses alarmes par scène est 1.45 en moyenne.



Suivi multi-caméras

collaboration avec J. Berclaz, R. Lengage et P. Fua

Suivi multi-caméras
Introduction

Étant donnés plusieurs flux vidéos, nous voulons suivre tous les individus qui passent dans la scène.

Séquence en intérieur

Suivi multi-caméras
Notations

Nous discrétisons la scène en un nombre fini de positions $N \simeq 1000$, et travaillons sur une séquence de $T \simeq 100$ “trames”.

- $\mathbf{I} = (I_1^1, \ldots, I_T^C)$ à valeurs dans $[0,1]^{3\times W\times H\times C\times T}$ les images provenant des caméras.
- $\mathbf{L}^m = (L_1^m, \ldots, L_T^m)$ à valeurs dans $\{1, \ldots, N\}^T$ la trajectoire de l'individu m.

Idéallement nous voudrions optimiser toutes les trajectoires

$$P(\mathbf{L}^1 = \lambda^1, \ldots, \mathbf{L}^M = \lambda^M \,|\, \mathbf{I}).$$

Suivi multi-caméras
Optimisation (1)

Nous optimisons les trajectoires successivement

$$\begin{aligned}
\hat{\lambda}^1 &= \arg\max_{\lambda} P(\mathbf{L}^1 = \lambda \,|\, \mathbf{I}),\\
\hat{\lambda}^2 &= \arg\max_{\lambda} P(\mathbf{L}^2 = \lambda \,|\, \mathbf{I}, \mathbf{L}^1 = \hat{\lambda}^1),\\
&\vdots\\
\hat{\lambda}^M &= \arg\max_{\lambda} P(\mathbf{L}^M = \lambda \,|\, \mathbf{I}, \mathbf{L}^1 = \hat{\lambda}^1, \ldots, \mathbf{L}^{M-1} = \hat{\lambda}^{M-1})
\end{aligned}$$

en modélisant le conditionnement comme une exclusion des emplacements déjà occupés

$$\begin{aligned}
&P(\mathbf{L}^m = \lambda \,|\, \mathbf{I}, \mathbf{L}^1 = \hat{\lambda}^1, \ldots, \mathbf{L}^{m-1} = \hat{\lambda}^{m-1})\\
&\qquad = P(\mathbf{L}^m = \lambda \,|\, \mathbf{I}, \forall u < m, \forall t, L_t^m \neq \hat{\lambda}_t^u)
\end{aligned}$$

Suivi multi-caméras
Optimisation (2)

Pour éviter les minima locaux, nous ordonnons les trajectoires selon la qualité de l'estimation passée afin que

- les individus faciles à suivre ne “volent” pas les trajectoires des autres,
- les individus difficiles à suivre soient traités en dernier, alors que le nombre de trajectoires possibles a été réduit.

Suivi multi-caméras
Modèle de trajectoire

Pour estimer

$$\arg\max_{\lambda} P(\mathbf{L}^m = \lambda \,|\, \mathbf{I}, \mathbf{L}^1 = \lambda^1, \ldots, \mathbf{L}^{m-1} = \lambda^{m-1})$$

sous un modèle de type Markov caché nous devons définir

- un modèle d'apparence $P(\mathbf{I}_t \,|\, L_t^m = n)$,
- un modèle de mouvement $P(L_t^m = n \,|\, L_{t-1}^m = \nu)$.

Notre modèle de mouvement est

$$P(L_t^m = n \,|\, L_{t-1}^m = \nu) = \begin{cases} \frac{1}{z}\, e^{-\rho\|n-\nu\|} & \text{si } \|n-\nu\| \leq c \\ 0 & \text{sinon} \end{cases}$$

Suivi multi-caméras
Modèle d'apparence

$\mathbf{I}_t$

$\mathbf{B}_t$

$\mathbf{T}_t(n)$

Nous notons $\mathbf{B}_t$ la partie qui a bougé, $\mathbf{T}_t(n)$ son contenu à la position n, et X_t^n l'occupation de la position n à l'instant t.

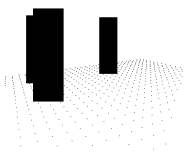
$$\overbrace{P(\mathbf{I}_t \,|\, L_t^m = n)}^{\text{Modèle d'apparence}} \propto P(L_t^m = n \,|\, \mathbf{I}_t)$$
$$= P(L_t^m = n \,|\, \mathbf{B}_t,\, \mathbf{T}_t(n))$$
$$= P(L_t^m = n \,|\, X_t^n = 1,\, \mathbf{B}_t,\, \mathbf{T}_t(n))\, P(X_t^n = 1 \,|\, \mathbf{B}_t,\, \mathbf{T}_t(n))$$
$$= \underbrace{P(L_t^m = n \,|\, X_t^n = 1,\, \mathbf{T}_t(n))}_{\text{Modèle de couleur}}\ \underbrace{P(X_t^n = 1 \,|\, \mathbf{B}_t)}_{\text{Occupation du sol}}$$

Suivi multi-caméras
Occupation du sol (1)

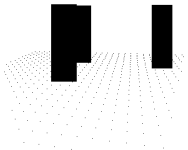
Nous modélisons $\mathbf{B}_t$ conditionnellement à $\mathbf{X}_t$ avec une hypothèse d'indépendance conditionnelle et une pseudo-distance Ψ

$$P(\mathbf{B}_t \,|\, \mathbf{X}_t) = \prod_c P(B_t^c \,|\, \mathbf{X}_t)$$
$$= \frac{1}{z} \prod_c e^{-\Psi(B_t^c, A_t^c)}$$

où A_t^c est une image construite en plaçant des silhouettes de tailles humaines aux positions occupées.



A_t^0



A_t^1

Suivi multi-caméras
Occupation du sol (2)

Nous cherchons la loi produit Q qui minimise la divergence de Kullback-Leibler avec la loi postérieure $P(\,\cdot \,|\, \mathbf{B})$. Avec $q_n = Q(X^n = 1)$, résoudre

$$\frac{\partial}{\partial q_n} KL(Q,\, P(\,\cdot \,|\, \mathbf{B})) = 0$$

mène à, en notant E_Q l'espérance sous $\mathbf{X} \sim Q$

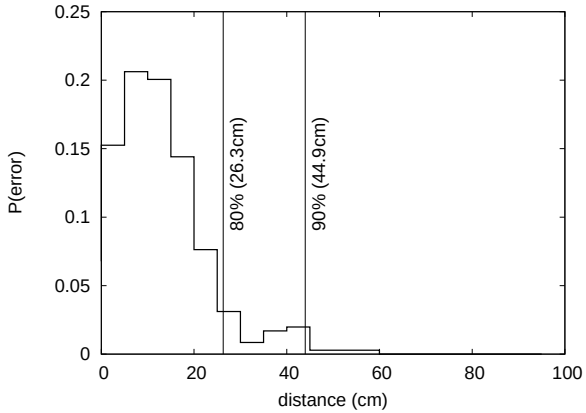
$$q_n = \frac{1}{1 + \exp\left(\lambda_n + \sum_c E_Q(\Psi(B^c, A^c) | X^n = 1) - E_Q(\Psi(B^c, A^c) | X^n = 0)\right)}$$

Nous résolvons itérativement une approximation de ce système.

deux caméras, quatre caméras, temp réel

Suivi multi-caméras
Résultats

Nous avons spécifié le résultat correct sur 100 “trames” choisies au hasard



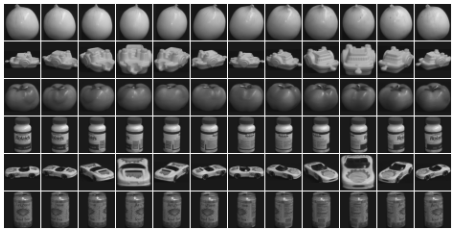
Séquence en extérieur

Apprentissage à partir d’un seul exemple

collaboration avec G. Blanchard

Apprentissage à partir d’un seul exemple
Introduction

Peut-on apprendre une invariance de haut niveau sur un grand nombre d’images d’un grand nombre d’objets



pour ensuite prédire si deux images montrent un même objet qui n’apparaît pas dans la base d’apprentissage ?

Apprentissage à partir d’un seul exemple
“Chopping”

Nous proposons de construire des fonctions

$$L_m : [0,1]^{32\times 32} \to \mathbb{R}$$

dont le signe est quasi-constant sur toutes les images de n’importe quel objet.

Nous comparons ensuite les valeurs de ces fonctions sur les deux images pour estimer si elles représentent le même objet.

Apprentissage à partir d'un seul exemple
Construction de L_m (1)

Les fonctionnelles L_m sont construites par apprentissage:

1. choisir au hasard un label booléen S_m invariant,
2. apprendre L_m:
   a. sélectionner K détecteurs de bords $X_{\nu(1)}, \ldots, X_{\nu(K)}$,
   b. entraîner un perceptron.

Apprentissage à partir d'un seul exemple
Construction de L_m (2)

Nous proposons une sélection de paramètres générique qui maximise l'information mutuelle conditionnelle

$$\begin{aligned} \nu(1) &= \arg\max_q \hat{I}(S_m ; X_q) \\ \forall k,\ 1 \leq k < K,\ \nu(k+1) &= \arg\max_q \min_{l \leq k} \hat{I}\left(S_m ; X_q \,|\, X_{\nu(l)}\right) \end{aligned}$$

Ce critère assure une sélection de paramètres *informatifs* et *complémentaires*.

Nous avons proposé une implantation "paresseuse" très efficace.

Apprentissage à partir d'un seul exemple
Modélisation (1)

Nous notons

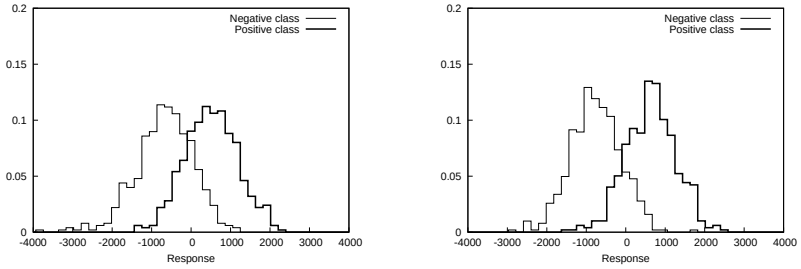
- I^1, I^2 à valeur dans $[0,1]^{32\times 32}$ les images,
- C^1, C^2 à valeurs dans $\{1, \ldots, N\}$ leurs classes,
- $\mathbf{S}^1$, $\mathbf{S}^2$ à valeurs dans $\{0, 1\}^M$ les labels binaires invariants,
- $\mathbf{L}^1$, $\mathbf{L}^2$ à valeurs dans $\mathbb{R}^M$ les réponses des prédicteurs.

Nous voulons tester si

$$P(C^1 = C^2 \,|\, \mathbf{L}^1, \mathbf{L}^2) \geq P(C^1 \neq C^2 \,|\, \mathbf{L}^1, \mathbf{L}^2)$$

Apprentissage à partir d'un seul exemple
Modélisation (2)

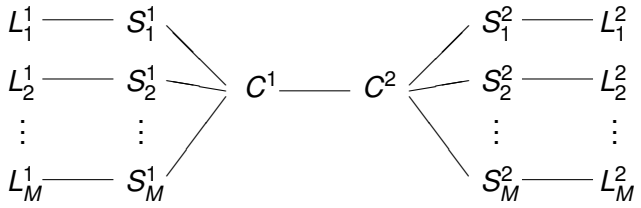
En utilisant d'autres images d'apprentissage, nous pouvons estimer les distributions empiriques de la réponse de L_m^i conditionnellement à S_m^i



que nous modélisons comme des densités gaussiennes. Ce modèle capture la qualité prédictive de L^m et nous permet d'estimer $P(S_m^i = 1 \,|\, L_m^i)$.

Apprentissage à partir d'un seul exemple
Modélisation (3)

Nous faisons une hypothèse d'indépendance conditionnelle des L_m^i étant donnés les S_m^i:



Apprentissage à partir d'un seul exemple
Prédiction (1)

Finalement, en notant

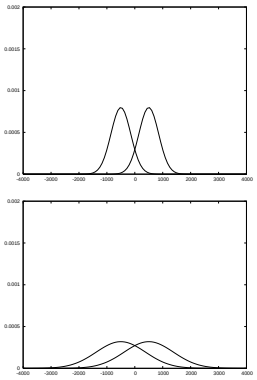
$$\alpha_m^i = P(S_m^i = 1 \mid L_m^i)$$

nous obtenons avec ce modèle

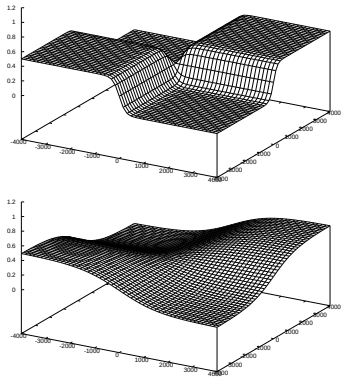
$$\log \frac{P(C^1 = C^2 \mid \mathbf{L}^1, \mathbf{L}^2)}{P(C^1 \neq C^2 \mid \mathbf{L}^1, \mathbf{L}^2)} = \sum_m \log\left(\alpha_m^1 \alpha_m^2 + (1-\alpha_m^1)(1-\alpha_m^2)\right) + \rho$$

Apprentissage à partir d'un seul exemple
Prédiction (2)

Exemple avec une seule fonction L_n

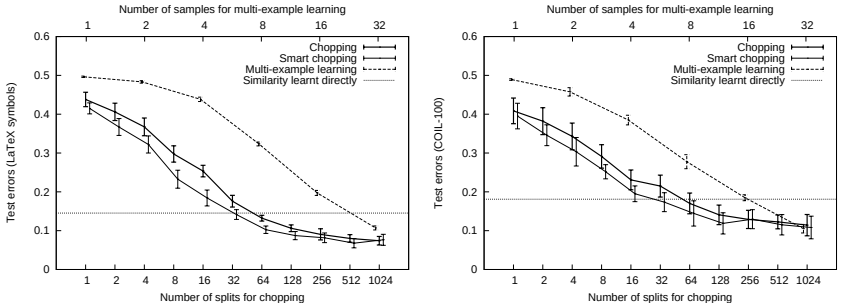


$P(L_n \mid S_n = 0)$ et $P(L_n \mid S_n = 1)$



$P(C^1 = C^2 \mid L_n^1, L_n^2)$

Apprentissage à partir d'un seul exemple
Résultats



Avec 1024 perceptrons et un seul exemple d'apprentissage les performances sont du même ordre qu'avec un seul perceptron entraîné avec 32 exemples positifs.

Conclusion

Les méthodes que nous avons développées reposent sur

- une combinaison de modèles discriminants et génératifs,
- des hypothèses d'indépendances conditionnelles,
- un contrôle fort du coût algorithmique.

Nos travaux actuels et futurs portent sur

- détection: objets à poses complexes (chats),
- suivi multi-caméras: modélisation fine du comportement,
- apprentissage avec un exemple: bornes de généralisation.

Merci!

François Fleuret
EPFL – CVLab
francois.fleuret@epfl.ch
http://cvlab.epfl.ch/~fleuret

Petits piétons