

Recherche d'informations visuelles

Nicolas Thome

Université Pierre et Marie Curie (UPMC)
Master Informatique - Spécialité DAC
Laboratoire d'Informatique de Paris 6 (LIP6)



Plan

1 Généralités sur la Recherche d'Informations

- Contexte
- Types de représentations visuelles
- Cas d'usages et défis de la RI

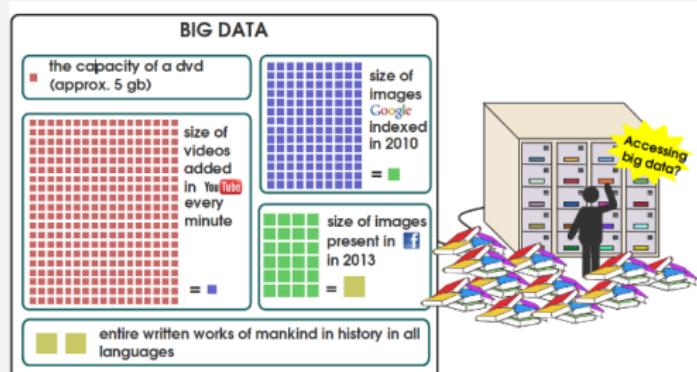
2 Index Visuels pour la RI

3 Recherche d'Informations visuelles structurées

Introduction à la Recherche d'Informations (RI)

Motivation

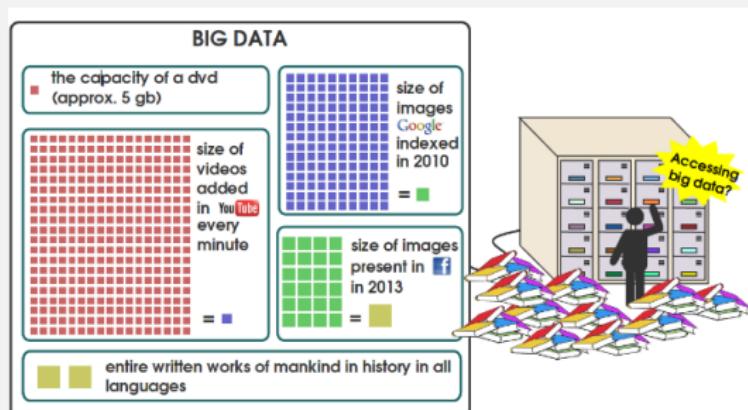
- Flot croissant d'information multimédia
- Augmentation de la capacité de stockage et de transmission
- Méga-banques de données = big data. Quelques chiffres :
 - En 2002, 5Mo TeraBytes de données créés
 - Google, 2005 : 8 milliard de pages indexées
 - Flickr : 540 millions d'images, google/yahoo image env 2 milliards
 - Vidéo : YouTube, INA, 700kh audio (radio) et 400kh video (télé)



Introduction à la Recherche d'Informations (RI)

Motivation

- MAIS... une « faible » capacité à traiter ces documents
 - Accès, sélection, comparaison, classement
 - Recherche d'information (Moteur de) : retrouver une image similaire
 - Fouille de données : structurer une grosse base en groupes homogènes
- ⇒ Nécessité de représentations des données et de méthodes de recherche efficaces



Introduction à la Recherche d'Informations Visuelles

Recherche d'Images : Motivation

① Recherche textuelle par mots clés

- Index textuel pour chaque image
- Recherche à partir de mots clés
- Problèmes
 - Disponibilité de l'information : coût de l'indexation
 - Définition du dictionnaire
 - Mise à jour

② Alternative : recherche visuelle par le contenu

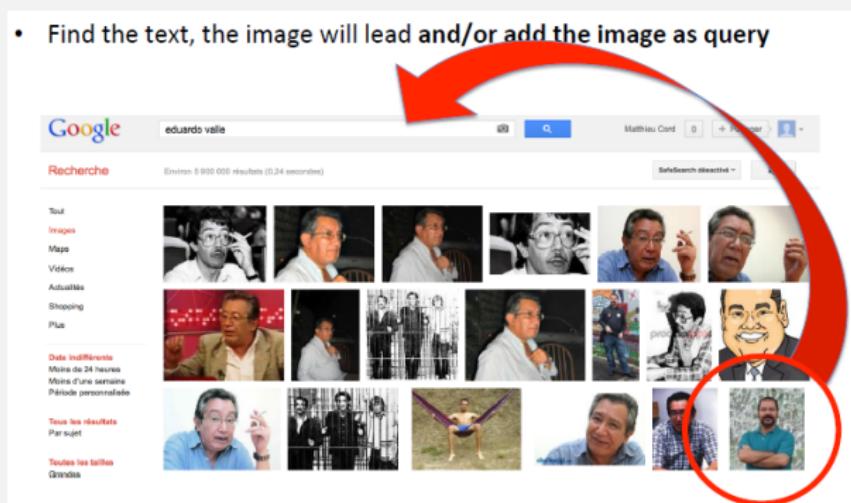
- Content-Based Image Retrieval (CBIR)
- Problème : comment décrire le contenu visuel d'une image ⇒ Ce cours

Introduction à la Recherche d'Informations Visuelles

Recherche d'Images : Changement (2012)

Problématique de la Recherche d'images par le contenu

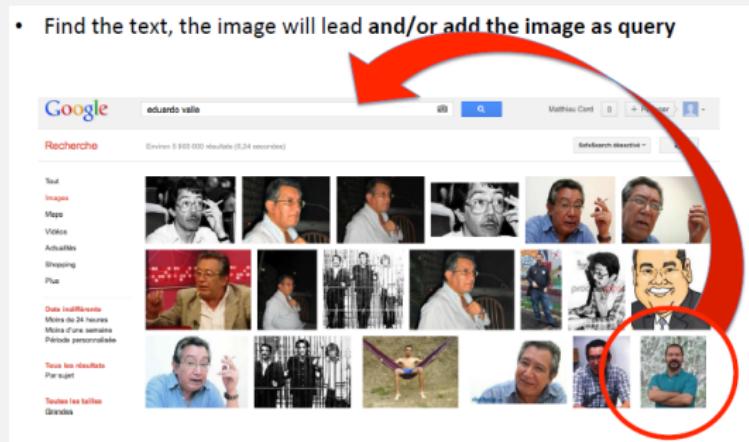
- Find the text, the image will lead and/or add the image as query



Introduction à la Recherche d'Informations Visuelles

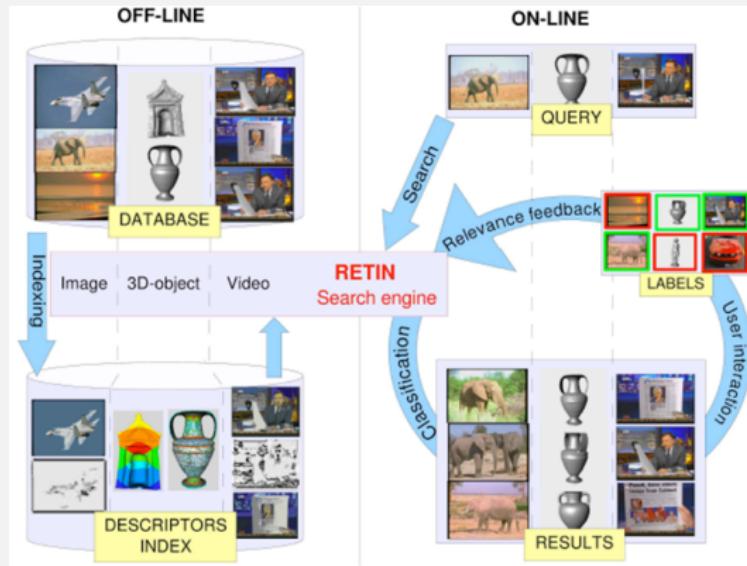
Problématique

- Quels sont les verrous dans les applications de recherche d'image par le contenu ?
- Problématique : comment représenter efficacement le contenu visuel d'une image ?



Introduction à la Recherche d'Informations Visuelles

Architecture des systèmes de CBIR



Plan

1 Généralités sur la Recherche d'Informations

- Contexte
- Types de représentations visuelles
- Cas d'usages et défis de la RI

2 Index Visuels pour la RI

3 Recherche d'Informations visuelles structurées

Types de représentations visuelles

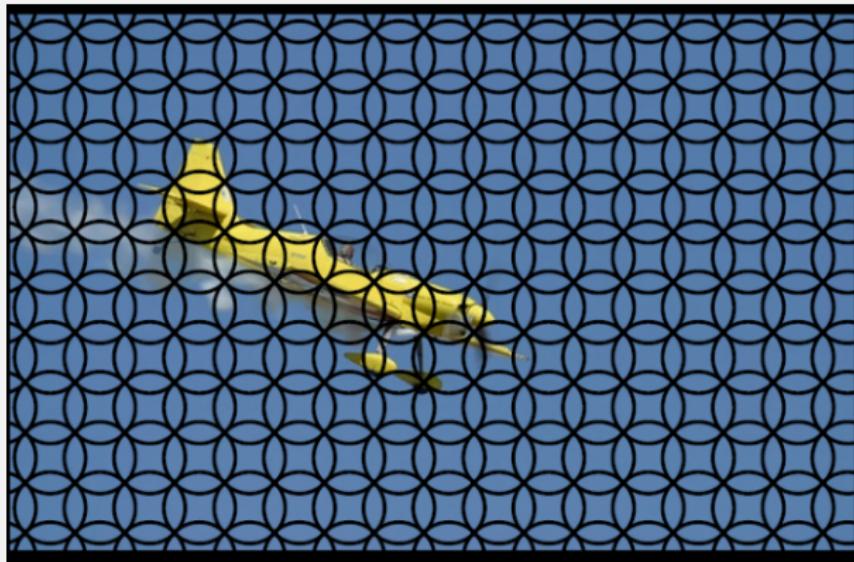
- Représentation "structurée" de chaque image de la base
- Terminologie :
 - Signature ⇒ relatif au contenu image
 - Index ⇒ relatif à la structure de stockage de l'information (informatique)
 - Flowchart du traitement : Primitives, Attributs, signature (descripteurs)



Types de représentations visuelles

Primitives

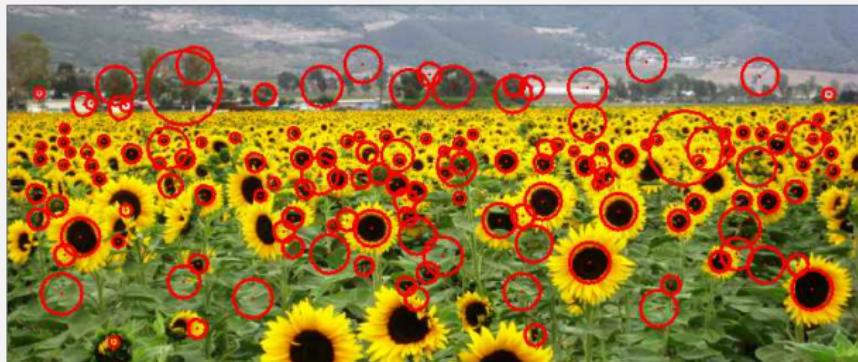
- Méthodes denses, non basées sur le contenu visuel :
 - Pixels, grille dense d'échantillonnage



Types de représentations visuelles

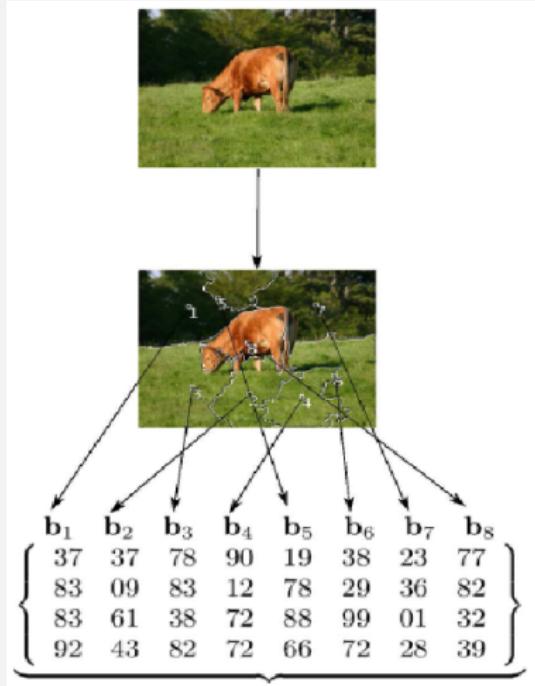
Primitives

- Méthodes éparses (sparse), basées sur le contenu visuel :
 - Points d'intérêt (image ou vidéo), segmentation en régions



Types de représentations visuelles

Attributs



- Couleur
- Forme
- Texture
- Gradients
- Mouvement
- etc

Types de représentations visuelles

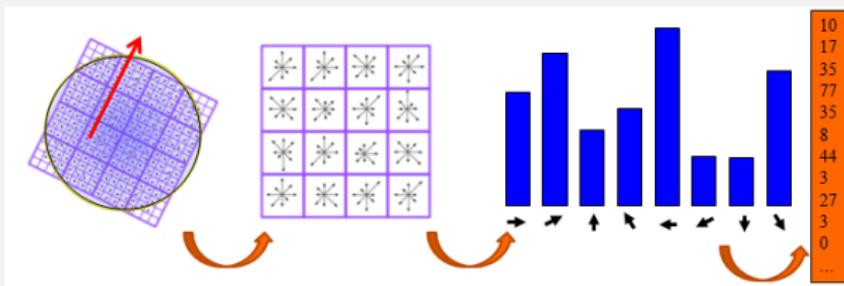
Signature = descripteurs

- Couleur : histogrammes RGB, HSV, etc
- Forme : Shape context, descripteurs de Fourier, etc
- Texture : Filtres de gabor etc
- Gradients : SIFT, SURF, HoG, etc
- Mouvement : HoF, STIP
- etc

Descripteurs

Un descripteur populaire : SIFT [Low04]

- Estimation de l'orientation dominante, recalage
- Découpage en (4x4) blocs, calcul d'histogrammes d'orientation de gradients



- \Rightarrow Vecteur de taille 128
- Post-traitements : seuillage valeur (> 0.2) et normalisation
- Invariances : géométrique (rotation), photométriques (changement affines d'illuminations)

Plan

1 Généralités sur la Recherche d'Informations

- Contexte
- Types de représentations visuelles
- Cas d'usages et défis de la RI

2 Index Visuels pour la RI

3 Recherche d'Informations visuelles structurées

Types de problèmes

Deux grands "paradigmes"

- ① Recherche de cibles ou identification d'images (un tableau, un objet)
 - Trouver le même document, mais (légèrement) déformé
- ② Recherche de catégories (une classe d'objets, un concept)
 - Trouver les documents qui contiennent le même concept
 - Problème du fossé sémantique

Premier Paradigme

Recherche de cibles ou identification d'images

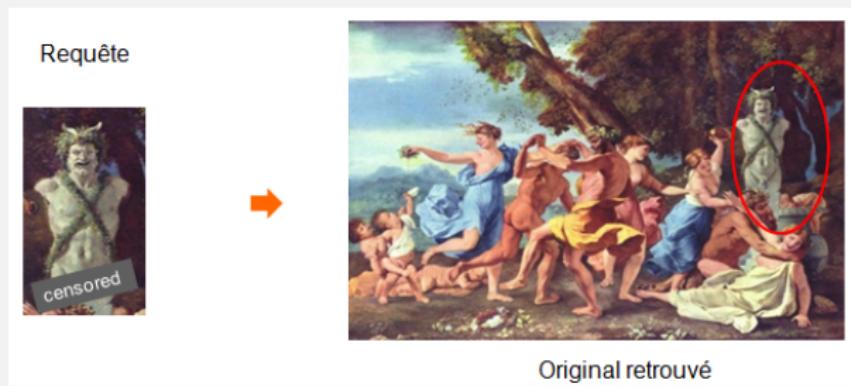


- Cas d'usage : rechercher de l'information à partir de photo mobiles (géo-localisées)

Premier Paradigme

Recherche de cibles ou identification d'images

- Exemple : recherche d'image en conservation du patrimoine
 - Images déformées ⇒ originales présentes dans le fond considéré ?
 - Recherche d'un objet particulier, d'un tableau



Premier Paradigme

Recherche de cibles ou identification d'images

Représentations visuelles : quelles prérogatives ?

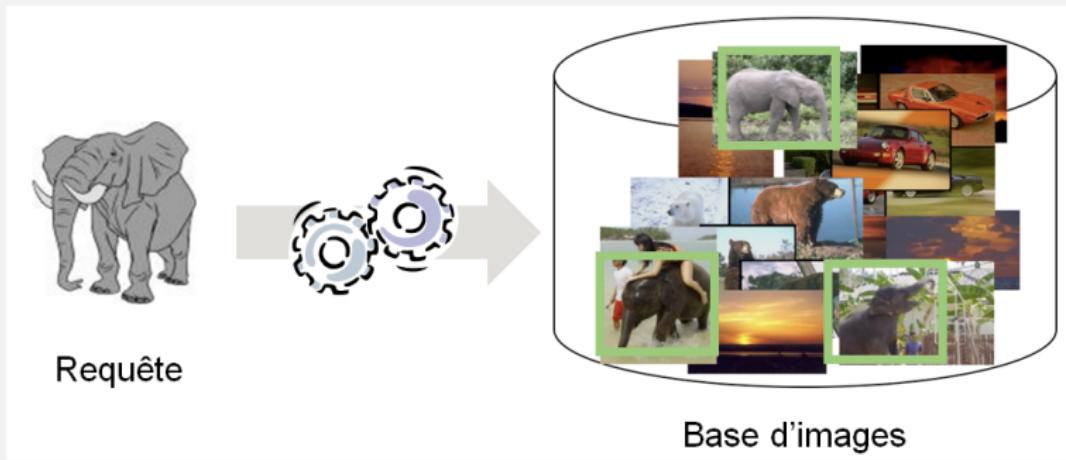
⇒ Dans ce contexte, on veut une représentation :

- Très discriminante : on cherche le même objet
- Robuste à des transformations
 - Géométriques
 - phométriques
 - Point de vue

2nd paradigme : recherche de catégories

Cas d'usage

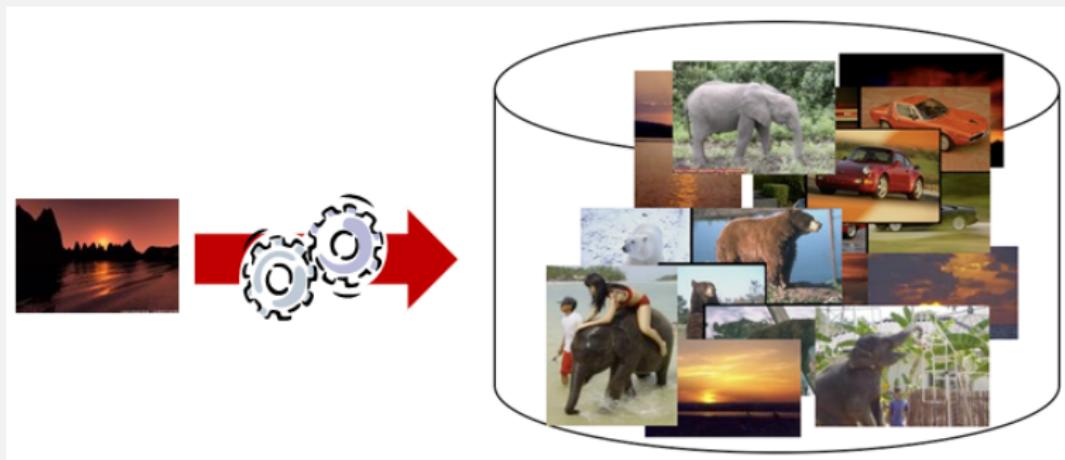
- Recherche de classes d'objets



2nd paradigme : recherche de catégories

Cas d'usage

- Recherche de catégories sémantiques
 - Recherche de type de scènes
 - caractéristiques dites haut niveau



2nd paradigme : recherche de catégories

Représentations visuelles : contraintes

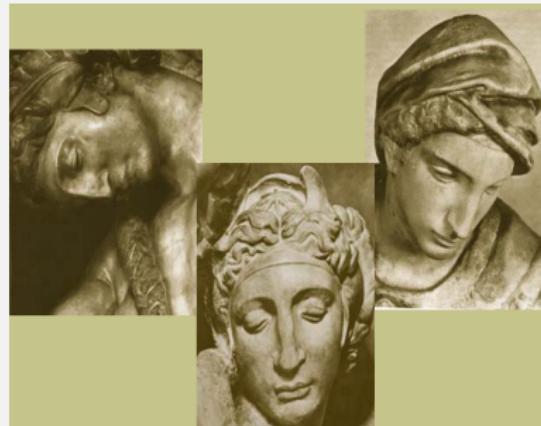
- Dans ce contexte, on veut une représentation
 - Discriminante : on cherche la même catégorie
 - Qui généralise aux autres images de la classe
 - Dans des conditions d'acquisition (géométriques, photométriques, point de vue) qui peuvent être très diverses

1er et 2nd paradigme : difficultés communes

Challenges



Variations Illumination



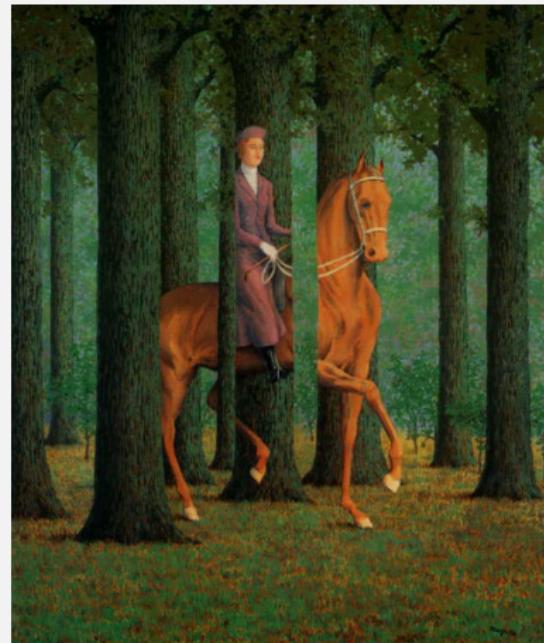
Variations point de vue

1er et 2nd paradigme : difficultés communes

Challenges



Variations échelle



Occultations

1er et 2nd paradigme : difficultés communes

Challenges



Objets Déformables



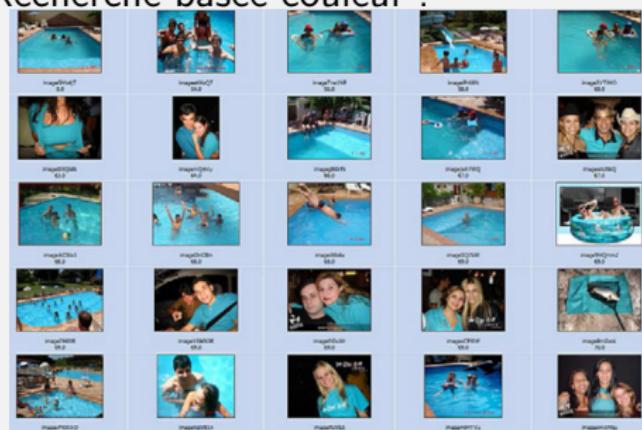
Background clutter

2nd paradigme : difficultés partiulières

Recherche de catégories

- Challenge : Problème du fossé sémantique
 - Descripteurs image : bas niveau (couleur, texture, forme, gradient)
 - Concepts à prédire : haut niveau (objets, scènes, concepts abstraits)
 - Principal défi : Combler le fossé sémantique

Recherche basée couleur :



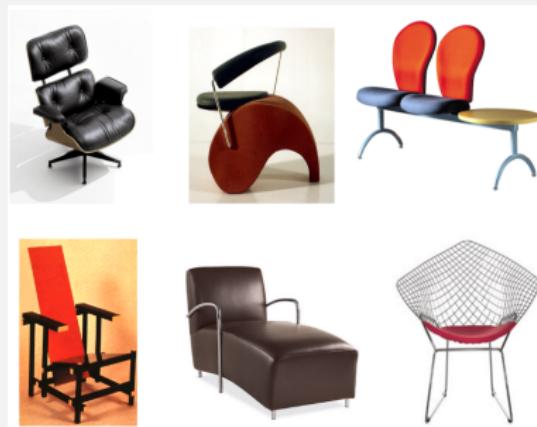
Requête :



2nd paradigme : difficultés partiulières

Recherche de catégories

- Challenge : variances intra et inter classes



2nd paradigme : difficultés partiulières

Recherche de catégories

- Challenge :
 - Taille de stockage et temps de calcul
 - Pas nécessairement de catégories connues ⇒ recherche en ligne, catégories mixtes,

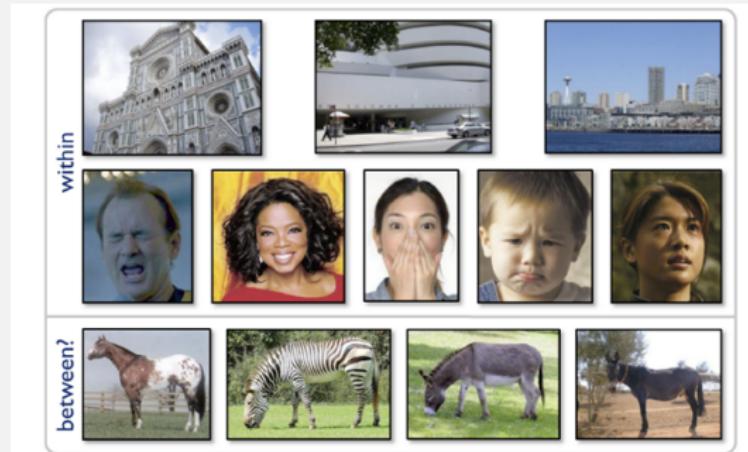


Figure 1.4: Within-class and between-class variability. Above: example images of buildings and faces illustrate the wide within-class variability in appearance of objects from particular categories. Below: depending on the application, horse, zebra, donkey, and mule could be considered (i) different classes, in which case their visual differences are important, or (ii) the same class (*Equidae*), in which case their visual similarities are important.

Plan

- 1 Généralités sur la Recherche d'Informations
- 2 Index Visuels pour la RI
- 3 Recherche d'Informations visuelles structurées

Quels indexs pour quels problèmes ?

Rappel : 2 paradigmes

- ① Recherche de cibles ou identification d'images (un tableau, un objet)
 - Trouver le même document, mais (légèrement) déformé
- ② Recherche de catégories (une classe d'objets, un concept) Trouver les documents qui contiennent le même concept Problème du fossé sémantique

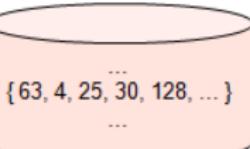
Problème 1 : cocktail gagnant \Rightarrow détecteur sparsePol + descripteurs discriminants et invariants type SIFT

- Local \Rightarrow intrinsèquement robustes aux occultations et au bruit (du fond)
- Challenge : recherche rapide de plus proches voisins dans les (très) grandes bases
- Vérification géométrique pour contraindre les correspondances

Paradigme 1 : identification d'images

Cocktail gagnant

Database of Images Find the Pol of Each Image Describe Each Pol Index all Descriptors



Offline

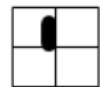
Query Image



Find the Pol



Describe Each Pol



Descriptor Matching

Refined Results



Geometric Consistency



Brute Results



Paradigme 2 : recherche de catégories

Cocktail gagnant

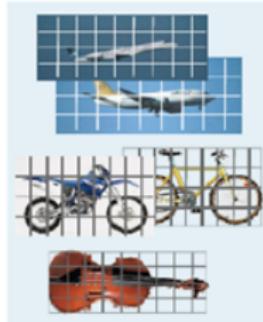
- Recherche de cible performante par mise en correspondance de descripteurs locaux
 - Mais ne passe pas l'échelle pour de très grandes bases
 - Nécessité d'approximation
- Catégorisation ≠ chercher des images "identiques" ⇒ généraliser
 - Modèle dominant dans les années 2000 : "Bag of Word" (BoW)
 - Approche issus de la communauté de recherche textuelle (sac de mots)



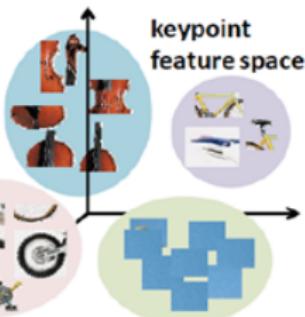
Modèle Bag of Words (BoW)

Pipeline

offline



clustering



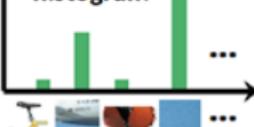
visual word vocabulary



online



BoW histogram



Modèle Bag of Words (BoW)

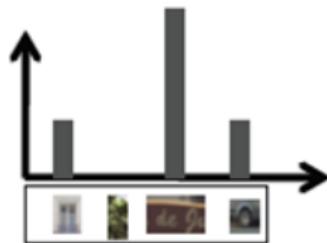
Améliorations

- BoW image [SZ03] ~ texte : histogramme de mots

Sac de descripteurs
(features)



BoW : histogramme
sur un dictionnaire



Modèle Bag of Words (BoW)

Améliorations

- Principale différence BoW image vs texte : notion de mot visuel
- Image : étape de clustering (ou apprentissage de dico, etc)
- Beaucoup de travaux sur l'encodage (soft, sparse, Fisher Vectors [PD07])
- Quelques travaux sur le pooling [dATC⁺13]

The diagram illustrates the Bag of Words (BoW) model and its improvements, specifically focusing on coding and pooling.

Left Matrix (Original BoW):

$$\mathbf{H} = c_m \begin{bmatrix} x_1 & x_j & x_N \\ \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \end{bmatrix} \Rightarrow g: \text{pooling}$$

Right Matrix (Improved Model):

$$\mathbf{H} = c_m \begin{bmatrix} x_1 & x_j & x_N \\ \alpha_{1,1} & \cdots & \alpha_{1,j} & \cdots & \alpha_{1,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{m,1} & \cdots & \alpha_{m,j} & \cdots & \alpha_{m,N} \\ \vdots & & \vdots & & \vdots \\ \alpha_{M,1} & \cdots & \alpha_{M,j} & \cdots & \alpha_{M,N} \end{bmatrix} \Rightarrow g: \text{pooling}$$

Annotations:

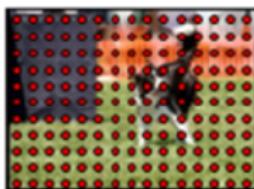
- f: coding** is indicated below the original matrix H.
- f: coding** is indicated below the improved matrix H.

Modèle Bag of Words (BoW)

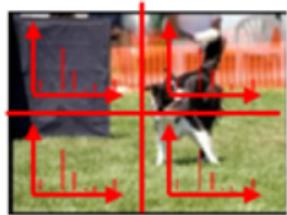
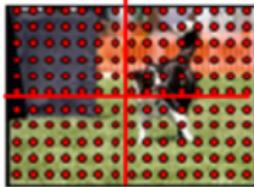
Améliorations : information spatiale [LSP06]



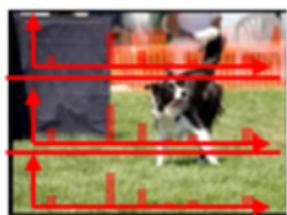
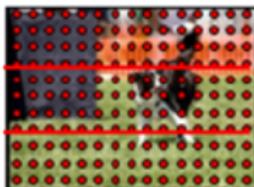
1x1

Vocabulary Assignment
(Bag-of-Words model)

2x2



3x1



Modèle Bag of Words (BoW)

En TME : représentation d'image

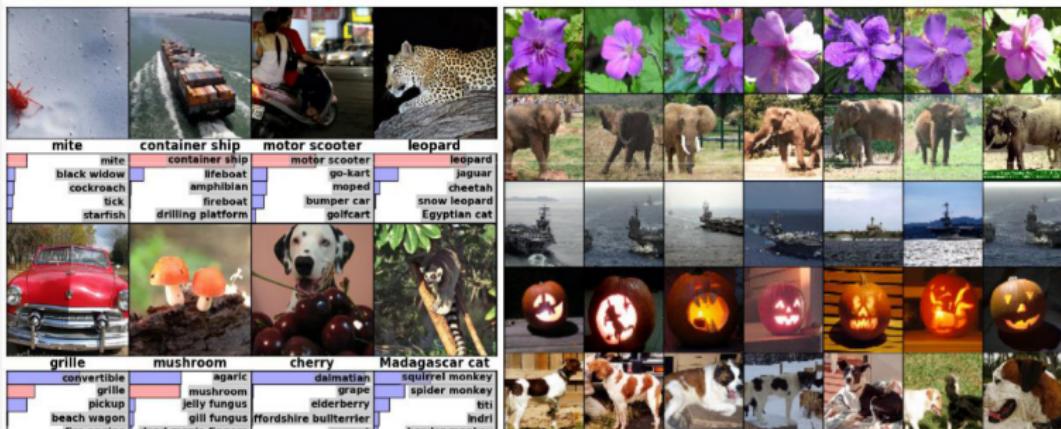
- Échantillonnage dense de points avec les informations :
 - Position (x,y)
 - Mot visuel associé (dictionnaire de taille 1000)
 - ID image
 - (Bounding box)
- Objectif 1 : générer les boW (codage dur, somme pooling, pas d'information spatiale)



Deep Learning

Recherche sémantique : au-delà du modèle BoW

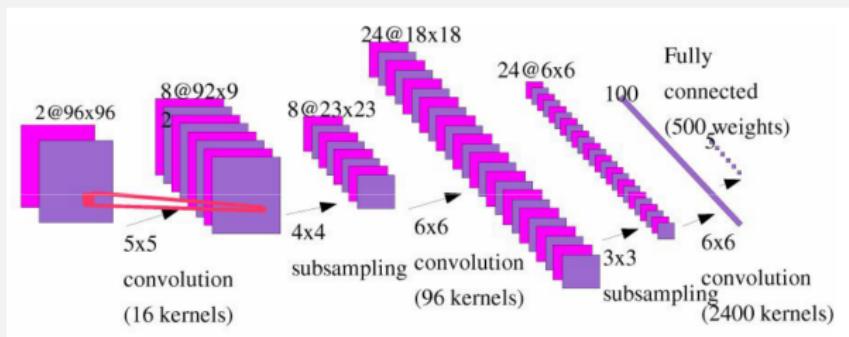
- BoW : approche dominante 2000 → 2012
- Récemment : depuis compétition ImageNet, conférence ECCV, octobre 2012
 - Tâche de classification très large échelle : 1000 classes, 1 million images entraînement
 - Approches basées deep learning [KSH12] surclassent BoW



Deep Learning

Recherche sémantique

- Réseaux de neurones convolutifs → gain très important par rapport aux méthodes traditionnelles (famille BoW), redéfinit l'état de l'art
 - Intérêt : combler le fossé sémantique en apprenant les représentations internes à partir des données
 - Réseaux pré-entraînés sur ImageNet ⇒ représentations pour différentes tâches (transfer learning)



Plan

- 1 Généralités sur la Recherche d'Informations
- 2 Index Visuels pour la RI
- 3 Recherche d'Informations visuelles structurées
 - Formulation de l'apprentissage
 - Étude de cas (instanciation)
 - Structural SVM : Optimisation

Apprentissage supervisé

Prédiction binaire vs prédiction structurée

- Apprentissage supervisé : optimiser les paramètres d'un modèle sur des données d'apprentissage \Rightarrow généraliser
 - Ex en RI : re-ranking des modèles de langage (TME 6)
- Prédiction binaire : données d'entrée $x \in \mathcal{X}$, fonction de prédiction $\hat{y}(x, w) = f_w(x) = (\text{par exemple}) = \text{sign}[\langle w, x \rangle (+b)]$
 - Prédiction $\hat{y}(x, w) \in \{-1; +1\}$
- Prédiction strcuturée : données d'entrée $x \in \mathcal{X}$, fonction de prédiction $\hat{y}(x, w) \in \mathcal{Y}$
 - \mathcal{Y} quelconque (pas scalaire) : vecteur, chaîne, graphe, etc

Apprentissage supervisé

Classification Binaire

- $x \in \mathbb{R}^d$, fonction de prédiction : $\hat{y}(x, w) = f_w(x) = \text{sign} [\langle w, x \rangle (+b)]$.
- Schéma d'apprentissage : optimiser w à partir d'un ensemble d'apprentissage $\{x_i\}_{i=1,\dots,N}$:

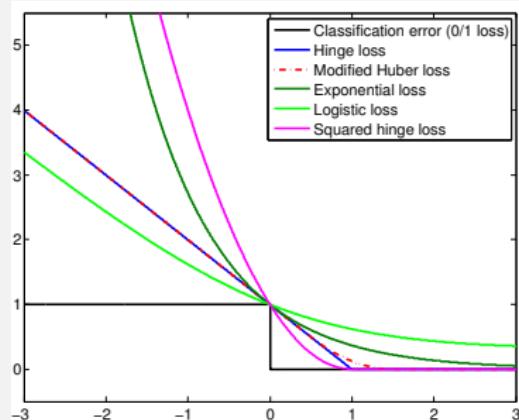
$$\mathcal{P}(w) = R(w) + \frac{C}{N} \sum_{i=1}^N \Delta[\hat{y}(x_i, w), y_i]$$

- Δ : fonction de coût entre y_i et $\hat{y}(x_i, w)$. Classif : "vrai" loss : 0-1 loss : $1_{\hat{y}y_i < 0}$
- MAIS 0-1 loss est NP hard à optimiser \Rightarrow surrogate loss

Classification Binaire

0-1 loss surrogate

- Surrogate loss : trouver une borne supérieure convexe $L_i(w)$ tq :
 $\Delta[\hat{y}(x_i, w), y_i] \leq L_i(w)$
- Ex : hinge loss : $[(w, x_i)y_i]_+ = \max[0, 1 - (w, x_i)y_i]$



$$\mathcal{P}(w) = R(w) + \frac{C}{N} \sum_{i=1}^N \max[0, 1 - (w, x_i)y_i]$$

Classification Binaire

Régularisation

- Régularisation avec norme ℓ_2 : $R(w) = \frac{1}{2} \|w\|^2$
- Cas du SVM : se prémunir contre le sur-apprentissage

$$\mathcal{P}(w) = \frac{1}{2} \|w\|^2 + \frac{C}{N} \sum_{i=1}^N \max [0, 1 - \langle w, x_i \rangle y_i]$$

- Bornes de généralisation
- Vision "smooth" de la régularisation : Cauchy-Schwarz :

$$\langle w, (x - x') \rangle \leq \|w\|^2 \|x - x'\|^2$$

- Une petite variation sur l'entrée x ne doit pas "trop" impacter le score $\langle w, x \rangle$. $\|w\|^2$ est une borne pour cette variation

Prédiction structurée (Structured output prediction)

Définition

- \mathcal{X} est l'espace d'entrée : arbitraire (pas nécessairement vectoriel, etc)
- \mathcal{Y} est l'espace de sortie structuré
 - Discret
 - Structuré : vecteur, chaîne, arbre, graphe, etc
- **Préliminaire** : définir une fonction décrivant la relation entre l'entrée x et la sortie y ("joint feature map") : $\Psi(x, y) \in \mathbb{R}^d$
- **Modèle** : apprendre un modèle de prédiction linéaire qui assigne le score $\langle w, \Psi(x, y) \rangle$ à chaque paire (x, y)
 - Extension à des fonctions non linéaire possible (kernelisation)
- Fonction de prédiction :

$$\hat{y}(x, w) = \arg \max_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle$$

Plan

- 1 Généralités sur la Recherche d'Informations
- 2 Index Visuels pour la RI
- 3 Recherche d'Informations visuelles structurées
 - Formulation de l'apprentissage
 - Étude de cas (instanciation)
 - Structural SVM : Optimisation

Prédiction structurée

Formulation de l'apprentissage

- On souhaite minimiser la fonction objectif suivante :

$$\mathcal{P}(w) = \frac{1}{2} \|w\|^2 \text{ s.t. :}$$

- $\forall x_i, i \in \{1, \dots, N\} \quad \forall y \neq y_i : \langle w, \Psi(x_i, y_i) \rangle \geq \langle w, \Psi(x_i, y) \rangle + 1$
- Marge de 1 comme pour les SVM binaires
- Variables "ressort" ξ_i (slack variables) :

$$\mathcal{P}(w) = \frac{1}{2} \|w\|^2 + \frac{C}{N(|\mathcal{Y}| - 1)} \sum_{i=1}^N \sum_{y \neq y_i} \xi_i \text{ s.t. :}$$

$$\forall x_i, \forall y \neq y_i : \langle w, \Psi(x_i, y_i) \rangle \geq \langle w, \Psi(x_i, y) \rangle + 1 - \xi_i$$

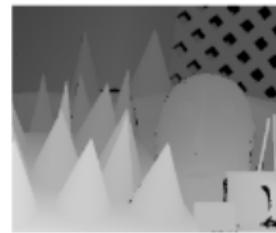
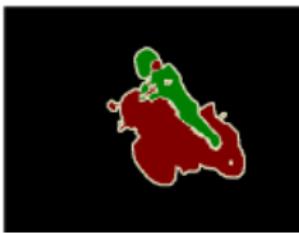
- Problème d'optimisation convexe

Prédiction structurée

Formulation de l'apprentissage

- $\forall x_i, \forall y \neq y_i : \langle w, \Psi(x_i, y_i) \rangle \geq \langle w, \Psi(x_i, y) \rangle + 1 - \xi_i$
- La marge de 1 est indépendante de la différence entre y and y_i ;
- Idée : utiliser $\Delta(y, y_i)$ pour refléter la connaissance *a priori* pour mesurer la différence entre y and y_i ;

Choix de la fonction de coût $\Delta(y, y_i)$: dépend de l'application



Prédiction structurée

Margin rescaling [TJHA05b]

- Plutôt que d'avoir une marge fixée à 1, fixer la marge $\Delta(y, y_i) : \forall x_i, \forall y \neq y_i : \langle w, \Psi(x_i, y_i) \rangle \geq \langle w, \Psi(x_i, y) \rangle + \Delta(y, y_i) - \xi_i$
- On peut montrer que la fonction objectif dans le cas du margin rescaling s'écrit :

$$\min_w \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^n \left[\max_{y \in \mathcal{Y}} [\Delta(y_i, y) + \langle \Psi(x_i, y), w \rangle] - \langle \Psi(x_i, y_i), w \rangle \right]$$

Prédiction structurée

Margin rescaling [TJHA05b]

- Autre façon de voir : la fonction objectif "idéale" serait :

$$P(w) = \min_w \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^n \Delta(y_i, \hat{y})$$

- MAIS optimiser $\Delta(y_i, \hat{y})$ est NP hard \Rightarrow surrogate loss

- Exercice : montrer que

$$\left[\max_{y \in \mathcal{Y}} [\Delta(y_i, y) + \langle \Psi(x_i, y), w \rangle] - \langle \Psi(x_i, y_i), w \rangle \right] \text{est une borne sup convexe à } \Delta(y_i, \hat{y})$$

Prédiction structurée

Slack rescaling [TJHA05b]

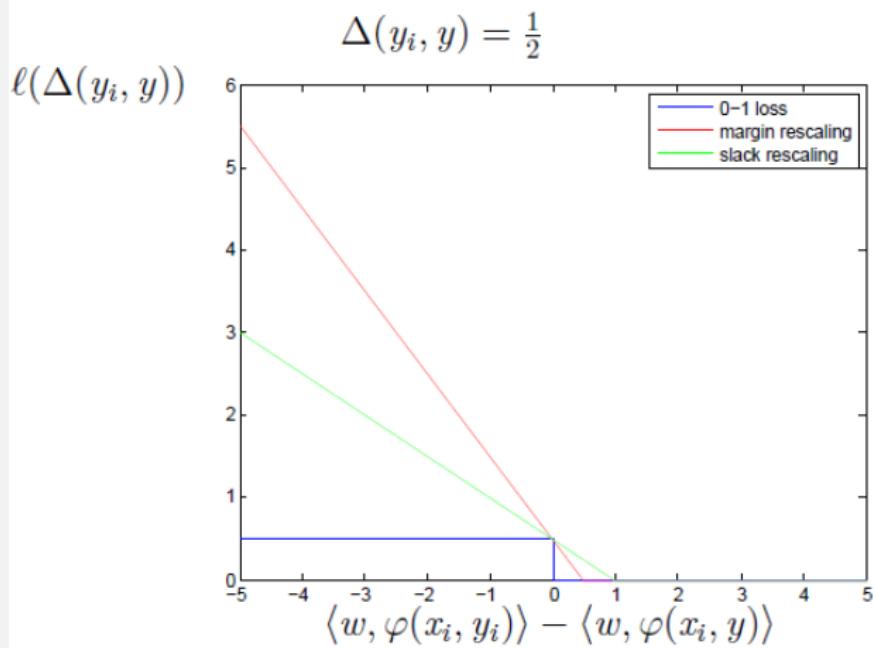
- $\forall x_i, \forall y \neq y_i : \langle w, \Psi(x_i, y_i) \rangle \geq \langle w, \Psi(x_i, y) \rangle - \frac{\xi_i}{\Delta(y, y_i)}$
- La fonction objectif devient

$$\min_w \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^n \max_{y \in \mathcal{Y}} [\Delta(y_i, y) (1 + \langle \Psi(x_i, y), w \rangle - \langle \Psi(x_i, y_i), w \rangle)] \quad (1)$$

- Comme dans le cas du margin rescaling, on peut montrer que $\max_{y \in \mathcal{Y}} [\Delta(y_i, y) (1 + \langle \Psi(x_i, y), w \rangle - \langle \Psi(x_i, y_i), w \rangle)]$ est une borne sup convexe à $\Delta(y_i, \hat{y})$

Prédiction structurée

Margin vs Slack rescaling



Plan

- 1 Généralités sur la Recherche d'Informations
- 2 Index Visuels pour la RI
- 3 Recherche d'Informations visuelles structurées
 - Formulation de l'apprentissage
 - **Étude de cas (instanciation)**
 - Structural SVM : Optimisation

SSVM : Instanciation

Rappel : ingrédient d'un problème d'apprentissage structuré

- Fonction de prédiction nécessite : $\Psi(x, y)$
- Apprentissage : $\Psi(x, y)$ et $\Delta(\mathbf{y}_i, \mathbf{y})$

Exemples classiques d'instantiations SSVM en vision

- Classification binaire
- Classification Multi-classes
- Classification hiérarchique
- Détection d'objets
- Ranking
- Estimation de pose
- Segmentation d'images (segmentation sémantique)
- Prédiction de séquences

SSVM : Instanciation

Classification Binaire

- $\Psi(x, y) = \frac{1}{2}y \cdot x, y \in \{-1; 1\}$
- $\Delta(y_i, y) = 1_{y_i \neq y}$
- Exercice : $\hat{y} = ?$, schéma d'apprentissage ?

SSVM : instantiation

Classification multi-classes

- $\mathcal{Y} = \{1, 2, \dots, K\}$, $\Delta(y, y') = \begin{cases} 1 & \text{for } y \neq y' \\ 0 & \text{otherwise.} \end{cases}$
- $\varphi(x, y) = (\llbracket y = 1 \rrbracket \Phi(x), \llbracket y = 2 \rrbracket \Phi(x), \dots, \llbracket y = K \rrbracket \Phi(x))$
 $= \Phi(x) e_y^\top \quad \text{with } e_y = y\text{-th unit vector}$

Solve:

$$\min_{w, \xi} \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^n \xi^i$$

subject to, for $i = 1, \dots, n$,

$$\langle w, \varphi(x^i, y^i) \rangle - \langle w, \varphi(x^i, y) \rangle \geq 1 - \xi^i \quad \text{for all } y \in \mathcal{Y} \setminus \{y^i\}.$$

Classification: MAP $f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$

Crammer-Singer Multiclass SVM

SSVM : instantiation

Classification multi-classes

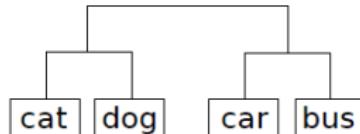
- Cas particulier de cette instantiation avec 2 classes ?
- Lien avec la classification binaire ?

SSVM : instantiation

Classification hiérarchique

Hierarchical Multiclass Classification

Loss function can reflect hierarchy:



$$\Delta(y, y') := \frac{1}{2}(\text{distance in tree})$$

$$\Delta(\text{cat}, \text{cat}) = 0, \quad \Delta(\text{cat}, \text{dog}) = 1, \quad \Delta(\text{cat}, \text{bus}) = 2, \quad \text{etc.}$$

Solve:

$$\min_{w, \xi} \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^n \xi^i$$

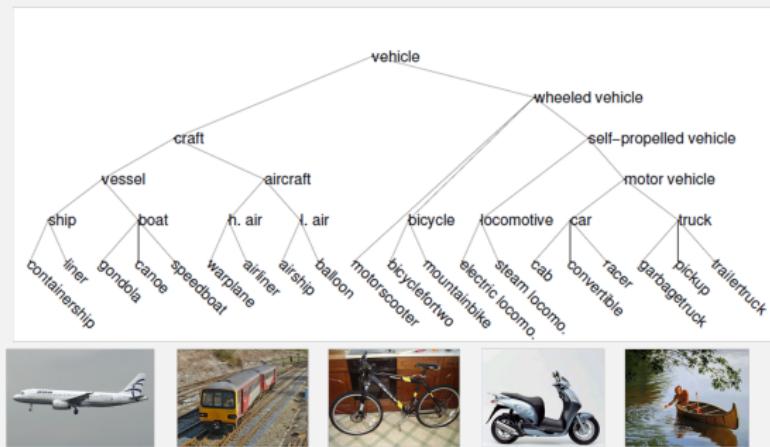
subject to, for $i = 1, \dots, n$,

$$\langle w, \varphi(x^i, y^i) \rangle - \langle w, \varphi(x^i, y) \rangle \geq \Delta(y^i, y) - \xi^i \quad \text{for all } y \in \mathcal{Y} \setminus \{y^i\}.$$

SSVM : instantiation

Classification hiérarchique

- Comment définir $\Delta(\mathbf{y}_i, \mathbf{y})$?
- Wordnet \Rightarrow distance sémantique entre classes
- En TME sur les classes de la base imageNet !



SSVM : instantiation

Détection d'objet : fonction de prédiction $\Rightarrow \psi(x, y) = ?$



output:
object position
(left, top
right, bottom)



- input space $\mathcal{X} = \{images\}$
- output space $\mathcal{Y} = \mathbb{R}^4$ bounding box coordinates
- scoring function $F(x, y) = w^\top \varphi(x, y)$ where $\varphi(x, y) = h(x|_y)$ is a feature vector for an image region, e.g. bag-of-visual-words.
- prediction function: $f : \mathcal{X} \rightarrow \mathcal{Y}$

$$f(x) := \operatorname{argmax}_{y \in \mathcal{Y}} F(x, y)$$

SSVM : instantiation

Détection d'objet : apprentissage $\Rightarrow \Delta(\mathbf{y}_i, \mathbf{y}) = ?$

\mathcal{X} images, $\mathcal{Y} = \{ \text{object bounding box} \} \subset \mathbb{R}^4$.

Training examples:



Goal: $f : \mathcal{X} \rightarrow \mathcal{Y}$



Loss function: area overlap $\Delta(y, y') = 1 - \frac{\text{area}(y \cap y')}{\text{area}(y \cup y')}$



SSVM : instantiation

Ranking : fonction de prédiction $\Rightarrow \Psi(x, y) = ?$

- Entrée \mathcal{X} : liste d'éléments, sortie \mathcal{Y} ordonnancement de ces éléments
 - $\mathbf{x} = (o_1, \dots, o_n)$
 - Ordonnancement peut être représenté par une matrice \mathbf{Y} tq :

$$y_{ij} = \begin{cases} +1 & \text{si } o_i <_y o_j \text{ (} o_i \text{ est classé avant } o_j \text{ dans la liste ordonnée)} \\ -1 & \text{si } o_i >_y o_j \text{ (} o_i \text{ est classé après } o_j \text{)} \end{cases}$$
 - Ranking feature map :

$$\Psi(x, y) = \sum_{i \in \Theta} \sum_{j \in \Theta} y_{ij} (\phi(o_i) - \phi(o_j))$$

- On apprend \mathbf{w} pour que les elts soient ordonnées comme le vrai ranking
- $\phi(o_i)$: description de o_i , e.g. $\phi(o_i) = b_i \in \mathbb{R}^d$ BoW ou feature deep (image)
- Question : comment déterminer $\hat{y}(x, w) = \arg \max_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle$?

SSVM : instantiation

Ranking : fonction de prédiction $\Rightarrow \Psi(x, y) = ?$

- Entrée $\mathcal{X} = \{q\}$: ensemble des requêtes q , sortie \mathcal{Y} : ordonnancement des éléments du corpus par rapport à chaque requête q .
 - Ranking feature map [YFRJ07] :

$$\langle \mathbf{w}; \Psi(x, y) \rangle = \sum_{i \in \mathcal{X}_q^+} \sum_{j \in \mathcal{X}_q^-} y_{ij} \langle \phi(q, o_i) - \phi(q, o_j); \mathbf{w} \rangle$$

- Ordonnancement peut être représenté par une matrice \mathbf{Y} tq :

$$y_{ij} = \begin{cases} +1 & \text{si } i <_y j \text{ (i est classé avant j dans la liste ordonnée / q)} \\ -1 & \text{si } i >_y j \text{ (i est classé après j)} \end{cases}$$
 - On apprend \mathbf{w} pour que les elts soient ordonnées comme le vrai ranking
 - $\phi(q, o_i)$: traduit la relation entre la requête et le document i , e.g. $\phi(q, o_i) = b_q - b_i$, où $b_q \in \mathbb{R}^d$ est la représentation de q (resp. b_i pour i)

SSVM : instantiation

Ranking et apprentissage de distance

$$\langle \mathbf{w}; \Psi(x, y) \rangle = \sum_{i \in \mathcal{X}_q^+} \sum_{j \in \mathcal{X}_q^-} y_{ij} \langle \phi(q, o_i) - \phi(q, o_j); \mathbf{w} \rangle$$

- Si on pose $\phi(q, o_i) = (b_q - b_i)(b_q - b_i)^T$ [ML10] :

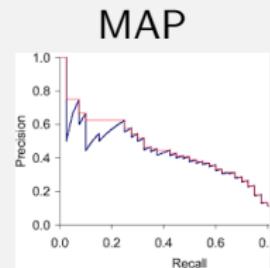
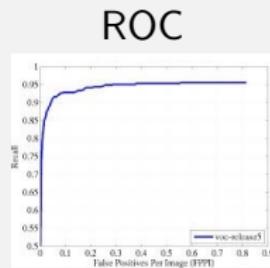
$$\langle \phi(q, o_i); \mathbf{w} \rangle = -(b_q - b_i)^T \mathbf{w} (b_q - b_i) = -d_{\mathbf{w}}(b_q; b_i) \quad (2)$$

- \mathbf{w} s'interprète alors comme la matrice de distance (Mahalanobis)
 - On veut que $d_{\mathbf{w}}(b_q; b_i) < d_{\mathbf{w}}(b_q; b_j)$ si $y_{ij} = +1$
- Souvent : imposer \mathbf{w} semi-définie positive (SDP) pour avoir une vraie métrique
- Modifie l'espace des solutions (cône SDP) lors de l'algorithme d'apprentissage

SSVM : instantiation

Ranking : apprentissage $\Rightarrow \Delta(\mathbf{y}_i, \mathbf{y}) = ?$

- $\Delta(\mathbf{y}_i, \mathbf{y}) = 1 - R(y)$, $R(y)$ mesure la qualité de ranking, e.g.
 - AUC, aire sous la courbe ROC (faux positifs vs vrais positifs)
 - $\Delta(\mathbf{y}_i, \mathbf{y}) = \frac{1}{n^2} \sum_{ij} y_{ij} \neq y_{ij}$
 - Ne tient pas compte de la position
 - Précision-at-k : ratio de doc pertinents retournés parmi les k premiers
 - Mean Average Precision (MAP), courbe sous la courbe rappel précision
 - etc
- Question : comment déterminer $\arg \max_{y \in \mathcal{Y}} [\langle w, \Psi(x, y) \rangle + \Delta(y, y_i)]$?

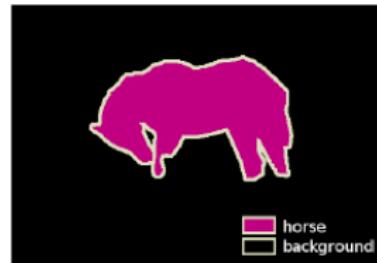


SSVM : instantiation

Segmentation sémantique d'images : prédiction $\Rightarrow \psi(x, y) = ?$



input: images



output: segmentation masks

- input space $\mathcal{X} = \{\text{images}\} \hat{=} [0, 255]^{3 \cdot M \cdot N}$
- output space $\mathcal{Y} = \{\text{segmentation masks}\} \hat{=} \{0, 1\}^{M \cdot N}$
- (structured output) prediction function: $f: \mathcal{X} \rightarrow \mathcal{Y}$

$$f(x) := \operatorname{argmin}_{y \in \mathcal{Y}} E(x, y)$$

- energy function $E(x, y) = \sum_i w_i^\top \varphi_u(x_i, y_i) + \sum_{i,j} w_{ij}^\top \varphi_p(y_i, y_j)$

SSVM : instantiation

Segmentation sémantique d'images : $\Delta(\mathbf{y}_i, \mathbf{y}) = ?$

- Données d'apprentissage : \mathcal{X} image, \mathcal{Y} masque de segmentation
- Loss utilisé : Hamming loss

$$\Delta(\mathbf{y}_i, \mathbf{y}) = \frac{1}{|V|} I(\mathbf{y}_i \neq \mathbf{y})$$
 - Compte le ratio de pixels mal étiquetés



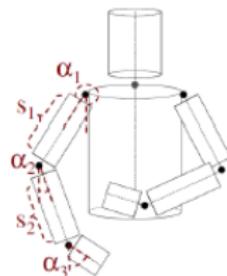
$t = 1: \hat{y} =$		$\phi(y^n) - \phi(\hat{y}):$ black +, white +, green -, blue -, gray -
$t = 2: \hat{y} =$		$\phi(y^n) - \phi(\hat{y}):$ black +, white +, green =, blue =, gray -
$t = 3: \hat{y} =$		$\phi(y^n) - \phi(\hat{y}):$ black =, white =, green -, blue -, gray -
$t = 4: \hat{y} =$		$\phi(y^n) - \phi(\hat{y}):$ black =, white =, green -, blue =, gray =

SSVM : instantiation

Estimation de pose : $\Delta(\mathbf{y}_i, \mathbf{y})$?



input: image



body model



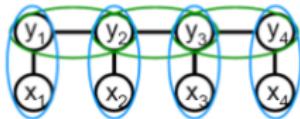
output: model fit

- input space $\mathcal{X} = \{\text{images}\}$
- output space $\mathcal{Y} = \{\text{positions}/\text{angle of } K \text{ body parts}\} \hat{=} \mathbb{R}^{3K}$.
- prediction function: $f : \mathcal{X} \rightarrow \mathcal{Y}$

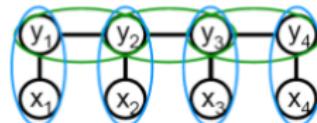
$$f(x) := \operatorname{argmin}_{y \in \mathcal{Y}} E(x, y)$$
- energy $E(x, y) = \sum_i w_i^\top \varphi_{\text{fit}}(x_i, y_i) + \sum_{i,j} w_{ij}^\top \varphi_{\text{pose}}(y_i, y_j)$

SSVM : instantiation

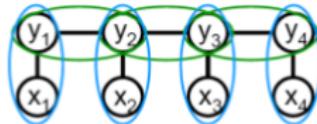
Prédiction de séquence



- Emissions (blue)
 - $f_e(x_i, y_i) = \langle w_e, \varphi_e(x_i, y_i) \rangle$
 - Can simply use the multi-class joint feature map for φ_e
- Transitions (green)
 - $f_t(y_i, y_{i+1}) = \langle w_t, \varphi_t(y_i, y_{i+1}) \rangle$
 - $\varphi_t(y_i, y_{i+1}) = \varphi_y(y_i) \otimes \varphi_y(y_{i+1})$ or $\begin{cases} [1 \ 0]^T & \text{if } y_i = y_{i+1} \\ [0 \ 1]^T & \text{if } y_i \neq y_{i+1} \end{cases}$



$$\begin{aligned}
 p(x, y) &\propto \prod_i e^{f_e(x_i, y_i)} \prod_i e^{f_t(y_i, y_{i+1})} \quad \text{for an HMM} \\
 f(x, y) &= \sum_i f_e(x_i, y_i) + \sum_i f_t(y_i, y_{i+1}) \\
 &= \langle w_e, \sum_i \varphi_e(x_i, y_i) \rangle + \langle w_t, \sum_i \varphi_t(y_i, y_{i+1}) \rangle
 \end{aligned}$$



$$\begin{aligned}
 w &= \begin{pmatrix} w_e \\ w_t \end{pmatrix} \\
 \varphi(x, y) &= \begin{pmatrix} \sum_i \varphi_e(x_i, y_i) \\ \sum_i \varphi_t(y_i, y_{i+1}) \end{pmatrix} \\
 f(x, y) &= \langle w, \varphi(x, y) \rangle
 \end{aligned}$$

Plan

- 1 Généralités sur la Recherche d'Informations
- 2 Index Visuels pour la RI
- 3 Recherche d'Informations visuelles structurées
 - Formulation de l'apprentissage
 - Étude de cas (instanciation)
 - Structural SVM : Optimisation

Structural SVM - Optimisation

Différentes formulations équivalentes :

Formulation 1 : n-slack

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{n} \sum_{i=1}^n \left[\max_{\mathbf{y} \in \mathcal{Y}} [\Delta(\mathbf{y}_i, \mathbf{y}) + \langle \Psi(\mathbf{x}_i, \mathbf{y}), \mathbf{w} \rangle] - \langle \Psi(\mathbf{x}_i, \mathbf{y}_i), \mathbf{w} \rangle \right]$$

ou :

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{n} \sum_{i=1}^n \xi_i$$

$$\forall i \in \{1, \dots, n\} \quad \max_{\mathbf{y} \in \mathcal{Y}} [\Delta(\mathbf{y}_i, \mathbf{y}) + \langle \Psi(\mathbf{x}_i, \mathbf{y}), \mathbf{w} \rangle] - \langle \Psi(\mathbf{x}_i, \mathbf{y}_i), \mathbf{w} \rangle \leq \xi_i$$

- n contraintes non-linéaires (à cause du max) 😞

Optimisation

Formulation 2 : n-slack

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{n} \sum_{i=1}^n \xi_i \quad (3)$$

$$\forall i \in \{1, \dots, n\}, \quad \forall \mathbf{y} \in \mathcal{Y}: \quad \langle \mathbf{w}; \delta\Psi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y}) \rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i, \quad \xi_i \geq 0 \quad (4)$$

avec $\delta\Psi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y}) = \Psi(\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \mathbf{y})$

- $n|\mathcal{Y}|$ contraintes linéaires ☺

Optimisation

Formulation 3 : one-slack

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C\xi$$

$$\forall (\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_n) \in \mathcal{Y} \times \dots \times \mathcal{Y}$$

$$\sum_{i=1}^n [\Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i) + \langle \Psi(\mathbf{x}_i, \hat{\mathbf{y}}_i), \mathbf{w} \rangle - \langle \Psi(\mathbf{x}_i, \mathbf{y}_i), \mathbf{w} \rangle] \leq n\xi \quad (5)$$

- Seulement 1 slack variable
- $|\mathcal{Y}|^n$ contraintes linéaires ☺

Optimisation

Différentes formulations : conclusion

- Formulation 1 a seulement n contraintes : peut être résolu directement
- Formulations 2 et 3 conduisent à un nombre de contraintes gigantesque. Exemple : n=100 images binaires de taille 16×16
 - Formulation 1 : 100 contraintes
 - Formulation 2 : $\sim 10^{79}$ contraintes
 - Formulation 3 : $\sim 10^{177}$ contraintes
- MAIS pour Formulations 2 & 3 : enforcer des contraintes "actives" est suffisant \Rightarrow working set training

Optimisation

Résolution de la formulation 1

Descente de (sous)-gradient

Algorithme 1 Algorithme d'apprentissage structuré

Entrées : Paires d'apprentissage $\{(x_i; y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i \in \{1; N\}}$, joint feature map $\Psi(x, y) \in \mathbb{R}^d$, loss $\Delta(y, y')$, régulariseur C , nombre d'itérations T , pas de gradient η_t .

Sortie : $w \Rightarrow$ fonction de prédiction $f_w(x) = \arg \max_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle$.

```

1:  $w \leftarrow \vec{0}$  // Initialisation
2: pour  $t = 1, \dots, T$  faire
3:   pour  $i = 1, \dots, N$  faire
4:      $(x_i; y_i) \leftarrow$  Sélection aléatoire d'une paire d'apprentissage
5:      $\hat{y} \leftarrow \arg \max_{y \in \mathcal{Y}} [\Delta(y, y_i) + \langle \Psi(x_i, y), w \rangle]$  // "Loss-augmented inference"
6:      $g_i = \Psi(x_i, \hat{y}) - \Psi(x_i, y_i)$  // Calcul du gradient
7:   fin pour
8:    $w \leftarrow w - \eta_t (w + \frac{C}{N} \sum_{i=1}^N g_i)$  // Mise à jour
9: fin pour
10: retourner w

```

Optimisation

Résolution de la formulation 1

Descente de (sous)-gradient stochastique : en TME !

Algorithme 2 Algorithme d'apprentissage structuré

Entrées : Paires d'apprentissage $\{(x_i; y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i \in \{1; N\}}$, joint feature map $\Psi(x, y) \in \mathbb{R}^d$, loss $\Delta(y, y')$, régulariseur C , nombre d'itérations T , pas de gradient η_t .

Sortie : $w \Rightarrow$ fonction de prédiction $f_w(x) = \arg \max_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle$.

```

1:  $w \leftarrow \vec{0}$  // Initialisation
2: pour  $t = 1, \dots, T$  faire
3:   pour  $i = 1, \dots, N$  faire
4:      $(x_i; y_i) \leftarrow$  Sélection aléatoire d'une paire d'apprentissage
5:      $\hat{y} \leftarrow \arg \max_{y \in \mathcal{Y}} [\Delta(y, y_i) + \langle \Psi(x_i, y), w \rangle]$  // "Loss-augmented inference"
6:      $g_i = \Psi(x_i, \hat{y}) - \Psi(x_i, y_i)$  // Calcul du gradient
7:      $w \leftarrow w - \eta_t(w + Cg_i)$  // Mise à jour
8:   fin pour
9: fin pour
10: retourner w

```

Optimisation

Résolution de la formulation 1 : cutting plane

$$L(w) =$$

$$\min_w \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^n \left[\max_{y \in \mathcal{Y}} [\Delta(y_i, y) + \langle \Psi(x_i, y), w \rangle] - \langle \Psi(x_i, y_i), w \rangle \right]$$

- $L(w)$: fonction convexe, non dérivable, avec une constante de Lipschitz bornée (i.e., ne varie pas trop vite).
 - Optimisation de ce genre de fonctions : intensivement étudiée en recherche opérationnelle
- Alternative à la descente de gradient : Bundle Method for Regularized Risk Minimization (BMRM)
 - Cas particulier des bundle method for regularised loss functions, qui sont une version stabilisée du cutting plane.

Optimisation

Résolution de la formulation 2 [TJHA05a]

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{n} \sum_{i=1}^n \xi_i \text{ s.t.}$$

$\forall i \in \{1, \dots, n\}, \forall \mathbf{y} \in \mathcal{Y} :$

$$\langle \mathbf{w}; \delta\Psi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y}) \rangle \geq \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i,$$

$$\xi_i \geq 0$$

$$\delta\Psi(\mathbf{x}_i, \mathbf{y}_i, \mathbf{y}) = \Psi(\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \mathbf{y})$$

Working Set S-SVM Training

input training pairs $\{(x^1, y^1), \dots, (x^n, y^n)\} \subset \mathcal{X} \times \mathcal{Y}$,
input feature map $\varphi(x, y)$, loss function $\Delta(y, y')$, regularizer C

```

1:  $S \leftarrow \emptyset$ 
2: repeat
3:    $(w, \xi) \leftarrow$  solution to QP only with constraints from  $S$ 
4:   for  $i=1, \dots, n$  do
5:      $\hat{y} \leftarrow \operatorname{argmax}_{y \in \mathcal{Y}} \Delta(y^i, y) + \langle w, \varphi(x^i, y) \rangle$ 
6:     if  $\hat{y} \neq y^i$  then
7:        $S \leftarrow S \cup \{(x^i, \hat{y})\}$ 
8:     end if
9:   end for
10:  until  $S$  doesn't change anymore.
```

output prediction function $f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$.

- ➊ Ligne 3 : résolue par (S)GD ou cutting plane
- ➋ Algorithme garanti de converger vers la même solution qu'avec l'ensemble des contraintes, à une précision ϵ
 - ➌ Pour une précision ϵ fixée, seulement $O(n/\epsilon^2)$ contraintes requises
(Tsochantaridis et.al, JMLR 2005)

Optimisation

Résolution de la formulation 3 [JFY09]

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C\xi \text{ s.t.}$$

$\forall (\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_n) \in \mathcal{Y} \times \dots \times \mathcal{Y} :$

$$\sum_{i=1}^n [\Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i) + \langle \delta\Psi(\mathbf{x}_i, \mathbf{y}_i, \hat{\mathbf{y}}), \mathbf{w} \rangle] \leq n\xi,$$

$$\xi_i \geq 0,$$

$$\delta\Psi(\mathbf{x}_i, \mathbf{y}_i, \hat{\mathbf{y}}) = \Psi(\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \hat{\mathbf{y}})$$

Working Set One-Slack S-SVM Training

input training pairs $\{(x^1, y^1), \dots, (x^n, y^n)\} \subset \mathcal{X} \times \mathcal{Y}$,
input feature map $\varphi(x, y)$, loss function $\Delta(y, y')$, regularizer C

- 1: $S \leftarrow \emptyset$
- 2: **repeat**
- 3: $(w, \xi) \leftarrow$ solution to QP only with constraints from S
- 4: **for** $i=1, \dots, n$ **do**
- 5: $\hat{y}^i \leftarrow \operatorname{argmax}_{y \in \mathcal{Y}} \Delta(y^i, y) + \langle w, \varphi(x^i, y) \rangle$
- 6: **end for**
- 7: $S \leftarrow S \cup \{(x^1, \dots, x^n), (\hat{y}^1, \dots, \hat{y}^n)\}$
- 8: **until** S doesn't change anymore.

output prediction function $f(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle w, \varphi(x, y) \rangle$.

Often faster convergence:

We add one strong constraint per iteration instead of n weak ones.

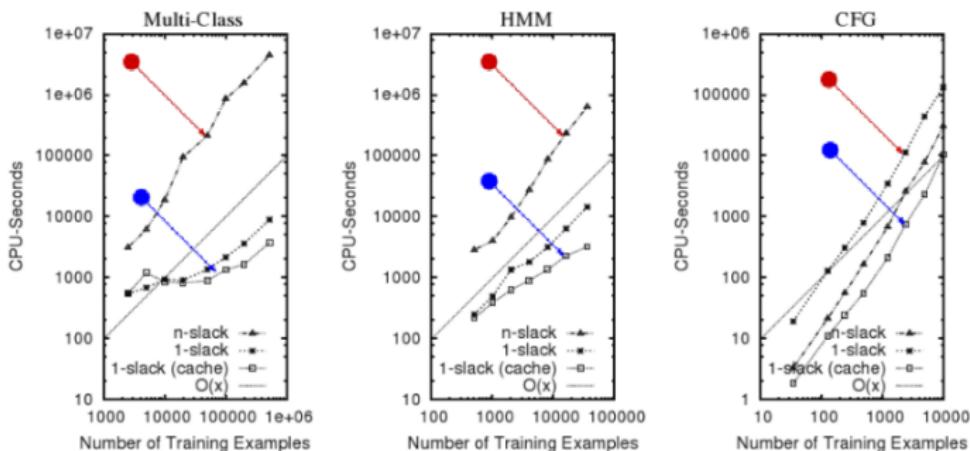
[Joachims, Finley, Yu: "Cutting-Plane Training of Structural SVMs", Machine Learning, 2009]

optimisation avec one-slack généralement plus rapide [JFY09]

Optimisation

Résolution de la formulation 3

Training Times ordinary S-SVM versus one-slack S-SVM



Observation 1: One-slack training is usually faster than n -slack.

Observation 2: Training structured models is nevertheless **slow**.

Figure: [Joachims, Finley, Yu: "Cutting-Plane Training of Structural SVMs", Machine Learning, 2009]

Optimisation

Inference et "loss-augmented" inference

Quelle que soit la formulation :

- Calcul de l'inférence $\hat{y}(x, w) = \arg \max_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle$ nécessaire en test
- Calcul du "loss-augmented inference" (ou "separation oracle")
 $\hat{y}(x, w) = \arg \max_{y \in \mathcal{Y}} (\langle w, \Psi(x, y) \rangle + \Delta(y, y_i))$ nécessaire en apprentissage
- Souvent le goulot d'étranglement
 - Pour des problèmes structurés avec $|\mathcal{Y}|$ grand, recherche exhaustive impossible !

Optimisation

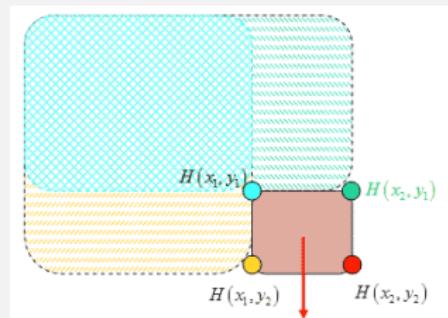
Inference et "loss-augmented" inference : étude de cas

- Classification multi-classes et hiérarchique
 - $|\mathcal{Y}| = K$ où K est le nombre de classes
 - Dans ce cas, il est envisageable de calculer l'inférence où le loss-augmented inference exhaustivement
- Détection d'objet : $|\mathcal{Y}|$: nombre de régions possibles
 - Approche classique : fenêtre glissante \Rightarrow très grand nb de régions candidates
 - Car on parcours à différentes positions, échelles, ratio (voire rotation)

Inference et "loss-augmented" inference : étude de cas

Détection d'objet

- Pour accélérer le calcul : "histogrammes intégrales"
 - Permet de calculer l'histogramme (BoW) en temps constant



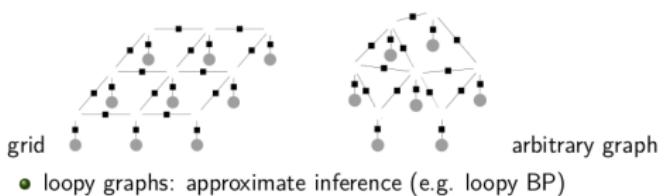
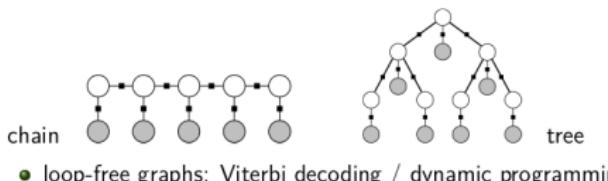
- Mais nécessite tjs l'évaluation de tous les produits scalaires
- autres méthodes : couper l'espace de recherche
 - stratégie branch and bound [BL08]

Inference et "loss-augmented" inference : étude de cas

Ranking

- $\Psi(x, y) = \sum_{ij} y_{ij} \langle \phi(o_i) - \phi(o_j); \mathbf{w} \rangle$
- Inférence : comment déterminer $\hat{y}(x, w) = \arg \max_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle$?
 - On peut montrer que \hat{y} revient à trier par ordre décroissant de $\langle \mathbf{w}; \phi(o_i) \rangle$
- "Loss-augmented" inference : $\arg \max_{y \in \mathcal{Y}} [\langle w, \Psi(x, y) + \Delta(y, y_i) \rangle]$?
 - Dépend de $\Delta(y, y_i)$
 - Des algorithmes existent pour AUC, MAP (plus compliqué) [YFRJ07]

Inference et "loss-augmented" inference : perspectives



- Segmentation :
 - inférence exacte dans des cas particuliers : structure chaîne, arbre, fonctions sub-modulaires pour les énergies
 - Sinon inférence approximée (e.g. loopy BP)
- Inference approximée : quelle garantie sur la convergence globale ??
Question ouverte...

Apprentissage structuré

Conclusion

- \oplus Formalisme général pour le problème, élégant, de nombreuses instantiations
- \oplus Problème d'optimisaton convexe
- \ominus Convexification (slack/margin rescaling) \Rightarrow modèle (trop) simple
- \oplus Des algorithmes efficaces pour résoudre
- \ominus Relativement lent pour des problèmes à forte strcuture
- \oplus SSVM proche des Conditionnal Random Field (CRF)
- \ominus Entrée sortie observés \Rightarrow pas d'apprentissage de paramètres cachés (représentations)

References I

-  Matthew B. Blaschko and Christoph H. Lampert, *Learning to localize objects with structured output regression*, Proceedings of the 10th European Conference on Computer Vision : Part I (Berlin, Heidelberg), ECCV '08, Springer-Verlag, 2008, pp. 2–15.
-  Sandra Eliza Fontes de Avila, Nicolas Thome, Matthieu Cord, Eduardo Valle, and Arnaldo de Albuquerque Araújo, *Pooling in image representation : The visual codeword point of view*, Computer Vision and Image Understanding **117** (2013), no. 5, 453–465.
-  T. Joachims, T. Finley, and Chun-Nam Yu, *Cutting-plane training of structural svms*, Machine Learning **77** (2009), no. 1, 27–59.
-  Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton, *Imagenet classification with deep convolutional neural networks*, NIPS (P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, eds.), 2012, pp. 1106–1114.
-  David G. Lowe, *Distinctive image features from scale-invariant keypoints*, Int. J. Comput. Vision **60** (2004), no. 2, 91–110.
-  Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, *Beyond bags of features : Spatial pyramid matching for recognizing natural scene categories*, CVPR, 2006.
-  Brian Mcfee and Gert Lanckriet, *Metric learning to rank*, In Proceedings of the 27th annual International Conference on Machine Learning (ICML), 2010.
-  Florent Perronnin and Christopher R. Dance, *Fisher kernels on visual vocabularies for image categorization*, CVPR, 2007.
-  J. Sivic and A. Zisserman, *Video google : A text retrieval approach to object matching in videos*, Proceedings of the International Conference on Computer Vision, vol. 2, October 2003, pp. 1470–1477.

References II



I. Tschantaridis, T. Joachims, T. Hofmann, and Y. Altun, *Large margin methods for structured and interdependent output variables*, Journal of Machine Learning Research (JMLR) **6** (2005), 1453–1484.



Ioannis Tschantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun, *Large margin methods for structured and interdependent output variables*, J. Mach. Learn. Res. **6** (2005), 1453–1484.



Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims, *A support vector method for optimizing average precision*, Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (New York, NY, USA), SIGIR '07, ACM, 2007, pp. 271–278.