

Reconnaissance de genre optimisée par GWO adaptatif

Benoit MARTIN¹, Julien MAROT², Salah BOURENNANE²

¹IntuiSense Technologies,
30 avenue du Château de Jouques
13420 Gemenos, France

²CNRS, Aix Marseille Université, Centrale Marseille, Inst. Fresnel,
52 av. Normandie 13397 Marseille, France

benoit.martin@intui-sense.com, julien.marot@fresnel.fr, salah.bourennane@fresnel.fr

Résumé – Nous proposons une méthode accélérée pour la classification de visages par genre. Nous estimons les paramètres optimaux pour la classification avec une version adaptative de l’algorithme d’optimisation du loup gris : nous accentuons l’aspect global ou local de la méthode. Simultanément, nous modifions itérativement la qualité de la base de données, pour affiner l’estimation tout en visant un faible temps de calcul. Sur la base FERET, nous atteignons un taux de bonne classification de 91.9%, et divisons par deux le temps de calcul par rapport à l’état-de-l’art.

Abstract – We propose an accelerated method for optimal gender classification. We estimate the optimal parameters for classification with an adaptive version of the grey wolf optimizer : either global or local optimization is emphasized. Simultaneously, we modify the quality of the database, to refine the parameter estimation while aiming at a low computational time. On the FERET face database, we reach a good classification rate of 91.9% and divide by two the computational time compared to existing methods.

1 Introduction

Pour classer par genre des images de visages, les modalités d’acquisition ont évolué [1], et il a été proposé d’extraire des motifs caractérisant la texture des visages. Par exemple, les motifs binaires locaux (local binary patterns ou LBP) [2]. L’idée principale de cet article est plutôt de profiter des récentes avancées dans le domaine de l’optimisation bio-inspirée pour estimer les paramètres d’un classifieur. Nous proposons d’adapter à ce problème une méthode récente d’intelligence en essaim.

Travaux antérieurs

Le classifieur ‘machines à vecteurs de support’ (support vector machines ou SVM) a montré son efficacité pour diverses applications. Cependant, il dépend de quelques paramètres et, sans connaissance *a priori*, la question de la valeur la plus adéquate des paramètres se pose toujours. La méthode habituelle et rapide consiste à tester quelques couples de paramètres, mais ce n’est ni rigoureux ni exhaustif. Depuis les premiers travaux concernant l’intelligence en essaim [3] (optimisation par essaim de particules ou PSO), celle-ci a été déclinée en plusieurs variantes [4] : mariage chez les abeilles, algorithme du ver luisant... Parmi celles-ci, celle qui aboutit au meilleur équilibre entre exploration ou recherche globale et exploitation ou recherche locale est la méthode du loup gris (grey wolf optimizer ou GWO). Proposé dans [4], GWO a été adapté à l’estimation des paramètres d’un SVM pour la classification d’images couleur [5] mais n’a jamais été appliqué à la classification de genre. GWO permet d’éviter les minima locaux mieux que PSO, mais le nombre d’agents de recherche et d’itérations restent élevés

[4].

Contributions

1) Premièrement, nous proposons un GWO adaptatif, qui permet d’allonger au choix la phase d’exploration ou la phase d’exploitation.

2) Deuxièmement, pour réduire le temps de calcul nécessaire à l’estimation des paramètres d’un SVM dans une application de classification d’images de visages, nous combinons trois versions de notre GWO adaptatif : une version globale, une version équilibrée, puis une version locale. Simultanément, nous adaptons les données de ‘réduites’ à ‘complètes’.

Plan

La section 2 pose notre problème de classification ; la section 3 décrit notre algorithme accéléré pour l’estimation des paramètres du SVM ; et la section 4 présente des résultats obtenus sur la base FERET [6].

2 Position du problème

Le processus de classification est appliqué à des motifs binaires locaux (local binary patterns ou LBP) [2] extraits des images de visages et qui caractérisent leur texture locale dans un voisinage de rayon r . Une image de LBP, puis son histogramme, sont calculés. Soit \mathbf{x} le vecteur contenant l’histogramme d’une image. Il s’agit d’un descripteur qui caractérise l’image, on suppose qu’il contient N composantes. La classification par genre consiste alors, à partir d’une base de données d’images divisée en base d’apprentissage et base de test, à iden-

tifier la catégorie (homme ou femme) à laquelle une image de la base de test appartient. Supposons que la base d'apprentissage contienne P images conduisant à P vecteurs histogrammes : $\{\mathbf{x}_i, y_i\}, i = 1, \dots, P$, où $\mathbf{x}_i \in \mathbb{R}^N$ et $y \in \{-1, +1\}$ est un label. La méthode SVM recherche un hyperplan optimal :

$$\min_{\mathbf{w}, b, \zeta_1, \dots, \zeta_P} \left[\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^P \zeta_i \right] \quad (1)$$

où $\zeta_i \geq 0$ sont des variables de relaxation, et \mathbf{w} est un vecteur de poids. C est un scalaire positif qui équilibre la marge entre les deux classes et le taux de mauvaise classification (TMC). Un vecteur test \mathbf{x} est classé selon :

$$f(\mathbf{x}) = \text{sign} \left(\sum_{i \in \mathbb{S}} \zeta_i y_i \mathcal{K}(\mathbf{x}_i, \mathbf{x}) + b \right) \quad (2)$$

où les ζ_i sont des multiplicateurs de Lagrange et \mathbb{S} est un sous-ensemble d'indices dans $\{1, 2, \dots, P\}$ qui définit les vecteurs de support. Dans l'équation (2), une fonction noyau est utilisée. Le noyau polynomial par exemple est défini par $\mathcal{K} = (\gamma \mathbf{x}_i^T \mathbf{x}_j + c_0)^d$. Le noyau linéaire en est un cas particulier où $d = 1$, $\gamma = 1$ et $c_0 = 0$. Sinon, le noyau polynomial dépend du degré d , de γ et de c_0 . Mais le noyau \mathcal{K} le plus souvent utilisé est le noyau Gaussien :

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \quad (3)$$

Les valeurs optimales de C et γ doivent être estimées pour toute application afin d'obtenir un résultat optimal en termes de taux de bonne classification. Pour cela, nous adaptons GWO [4], où quatre types de loups sont utilisés : trois leaders α, β, δ , et les subalternes ω . La méthode d'optimisation GWO modélise le comportement des loups, et assimile la recherche du minimum d'une fonction coût à la stratégie de chasse d'une meute de loups. La traque de la proie correspond à l'exploration ou recherche globale, l'attaque à l'exploitation ou recherche locale. Soit $\vec{\mathbf{X}}(t)$ la position d'un loup à l'itération t . Cette position est mise à jour en tenant compte des positions des trois 'leaders' (α, β, δ) qui obtiennent les meilleurs scores :

$$\vec{\mathbf{X}}(t+1) = \frac{1}{3} (\vec{\mathbf{X}}_\alpha + \vec{\mathbf{X}}_\beta + \vec{\mathbf{X}}_\delta) \quad (4)$$

La position des loups α, β, δ est calculée ainsi :

$$\vec{\mathbf{X}}_l = \vec{\mathbf{X}}_l - \vec{\mathbf{A}}_l \cdot (\vec{\mathbf{D}}_l), \quad l = \alpha, \beta, \text{ or } \delta \quad (5)$$

avec :

$$\vec{\mathbf{D}}_l = |\vec{\mathbf{C}}_l \cdot \vec{\mathbf{X}}_l - \vec{\mathbf{X}}(t)|, \quad l = \alpha, \beta, \text{ or } \delta \quad (6)$$

Les vecteurs $\vec{\mathbf{A}}$ et $\vec{\mathbf{C}}$ sont calculés par $\vec{\mathbf{A}} = 2\vec{\mathbf{a}} \cdot \vec{\mathbf{r}}_1 - \vec{\mathbf{a}}$ et $\vec{\mathbf{C}} = 2 \cdot \vec{\mathbf{r}}_2$. Le paramètre a , dans [4], décroît régulièrement de 2 à 0 entre $t = 1$ et $t = T$: $a = 2(1 - \frac{t}{T})$. Durant la chasse, la phase d'exploration court jusqu'à ce que la condition $|\vec{\mathbf{A}}| \geq 1$ soit respectée : les loups sont forcés à s'éloigner les uns des autres. Ensuite commence la phase d'exploitation, dès que $|\vec{\mathbf{A}}| < 1$: les loups sont alors forcés d'attaquer la proie. Par

conséquent, un paramètre clé pour équilibrer les phases d'exploration et d'exploitation est le paramètre a . La recherche globale s'arrête à $a = 1$. Commence alors la recherche locale, jusqu'à $a = 0$. Dans [7], un GWO modifié (mGWO) est implémenté avec un modèle quadratique pour a : $a = 2(1 - \frac{t^2}{T^2})$. Cela conduit à une phase d'exploration plus longue, au détriment de la phase d'exploitation.

En section 3, nous généralisons ce modèle.

3 GWO adaptatif et classification

Un inconvénient majeur dans les méthodes d'optimisation existantes est le nombre élevé d'itérations et de nombre d'agents requis pour qu'elles convergent. Dans [8] par exemple, pour 250 itérations et 50 agents sont utilisés pour GWO et PSO. Nous proposons une version généralisée du paramètre a :

$$a = 2(1 - \frac{t^\eta}{T^\eta}) \quad (7)$$

où η est un scalaire $\in \mathbb{R}_+^*$. Le nombre d'itérations dédiées à l'exploration est :

$$t_{\text{exploration}} = \frac{T}{2^{1/\eta}} \quad (8)$$

Premièrement, nous proposons une version 'locale' de GWO. Pour cela nous proposons de choisir $\eta < 1$. Par exemple pour $\eta = 0.5$, $t_{\text{exploration}} = \frac{T}{4}$.

Nous proposons de faire varier le nombre de leaders : l'équation (4) est modifiée comme suit :

$$\vec{\mathbf{X}}(t+1) = \frac{1}{N_l} \left(\sum_{l=1}^{N_l} \vec{\mathbf{X}}_l \right) \quad (9)$$

où N_l est le nombre de leaders, et $\vec{\mathbf{X}}_1, \vec{\mathbf{X}}_2, \dots, \vec{\mathbf{X}}_{N_l}$ sont les N_l loups correspondant aux meilleurs scores. nous appelons cette version avec η et N_l variables le 'GWO adaptatif'.

Deuxièmement, pour la première fois dans cet article, nous adaptons GWO à la classification d'images de visages. Dans cette application $\vec{\mathbf{X}}$ contient les paramètres pour SVM : $\vec{\mathbf{X}} = [\gamma, C]^T$. Pour réduire le temps de calcul, nous nous contrainsons à choisir des valeurs faibles pour le nombre d'agents et d'itérations, et nous modifions les caractéristiques de la base de données, pour affiner l'estimation des valeurs de paramètres.

Une base de données 'réduite' contient une valeur faible P d'images sous-échantillonnées à une taille S . L'algorithme 1 est composé de trois étapes, et consiste à appliquer trois fois GWO en l'adaptant : au début de l'algorithme L'algorithme 1 nous souhaitons privilégier la phase d'exploration sur une base de données réduite. Pour cela, nous choisissons une valeur élevée pour η (voir Eq. (7)). Nous utilisons un nombre réduit d'images de faible taille, afin que cette phase soit la plus rapide possible. Lors d'une étape intermédiaire, la phase d'exploration est moins longue (avec $\eta = 2$), et un nombre d'images deux fois plus grand est utilisé pour la base d'apprentissage.

Finalement, une dernière étape privilégie la phase d'exploitation (avec $\eta = \frac{1}{2}$) qui permet d'affiner l'estimation des paramètres sur une base de données comprenant un grand nombre d'images de grande taille et donc de bonne résolution.

Algorithm 1 Pseudo-code : estimation accélérée des paramètres de classification par loup gris adaptatif.

Entrées : Composantes LBP avec une valeur de rayon $r = 1$ obtenues à partir des images des bases d'entraînement et de test pour calculer les LBP. **Fonction objectif** : le taux de mauvaise classification.

Paramètres image : taille S , nombre d'images P pour chaque classe dans la base d'entraînement.

Paramètres pour l'optimisation : nombre de leaders N_l , η , nombre de loups N_a , nombre d'itérations T .

Appliquer trois fois GWO sur les LBP avec les valeurs de paramètres suivantes :

1. $S = 128, P = 100; N_l = 6, \eta = 8, N_a = 25, T = 5$.
Renvoyer $\vec{X}_{\alpha 1}$.
2. $S = 128, P = 200; N_l = 4, \eta = 2, N_a = 20, T = 5$, et $\vec{X}_{\alpha 1}$ comme position d'initialisation de l'un des loups.
Renvoyer $\vec{X}_{\alpha 2}$.
3. $S = 256, P = 400; N_l = 3, \eta = \frac{1}{2}, N_a = 15, T = 5$, et $\vec{X}_{\alpha 2}$ comme position d'initialisation de l'un des loups.

Sortie : $\vec{X}_\alpha = [\hat{\gamma}, \hat{C}]^T$ contenant les valeurs optimales des paramètres SVM pour la classification par genre.

La section 4 justifie la méthode en trois étapes que nous proposons et compare les résultats qu'elle obtient à ceux de l'état-de-l'art.

4 Résultats

Les expériences sont réalisées avec un PC à 3GHz sous Windows, sur la base d'images FERET (voir quelques exemples en Fig. 1) [6]. Sauf indication contraire les images sont de taille $S \times S$ avec $S = 256$; les bases d'apprentissage et de test contiennent respectivement $P = 200$ images, et 500 images pour chaque genre; le rayon pour LBP est $r = 1$ et le noyau pour SVM est Gaussien. Les valeurs $r = 2$ ou 3, et d'autres noyaux conduisent à des valeurs de TMC plus élevés. PSO est appliquée avec des constantes d'accélération valant 2, et une inertie variant de 0.9 à 0.4 [3].

4.1 Paramètres du SVM et résultats de classification

La Table 1 fournit quelques résultats de classification, et montre que les résultats en termes de TMC mais aussi de temps de calcul dépendent fortement des paramètres γ et C .

Le meilleur couple de paramètres, $(\gamma, C) = (10^{-6}, 0.58 \cdot 10^3)$,

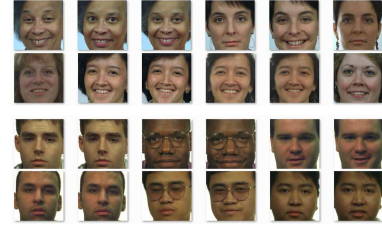


FIGURE 1 – Base d'images FERET : exemples

| (γ, C) | temps (ms) | 100-TMC |
|------------------------------|------------|---------|
| $(10^{-6}, 0.8 \cdot 10^3)$ | 4641 | 88.4% |
| $(10^{-3}, 0.8 \cdot 10^3)$ | 9219 | 50.0% |
| $(10^{-6}, 1)$ | 5141 | 83.7% |
| $(10^{-3}, 1)$ | 9119 | 50.0% |
| $(10^{-6}, 0.58 \cdot 10^3)$ | 4640 | 88.6% |
| $(10^{-3}, 0.58 \cdot 10^3)$ | 9234 | 50.0% |

TABLE 1 – Classification vs. paramètres SVM

conduit, pour différentes valeurs de S et P , aux performances données en Table 2.

| S | P | temps (ms) | 100-TMC |
|-----|-----|------------|---------|
| 32 | 100 | 2637 | 75.6% |
| 64 | 100 | 2122 | 83.0% |
| 128 | 100 | 1451 | 86.1% |
| 128 | 200 | 2948 | 88.3% |
| 256 | 400 | 6567 | 91.0% |

TABLE 2 – Classification vs paramètres image

La Table 2 montre que trop réduire la taille des images ne diminue ni le TMC ni le temps de calcul. En revanche, il est montré que choisir les valeurs $S = 128$ et $P = 100$ réduit de manière significative le temps de calcul, tout en préservant le taux de bonne classification.

4.2 Paramètres de classification optimaux pour différents ensembles d'images

La Table 3 présente les meilleurs paramètres γ et C obtenus, et les résultats de classification correspondants pour plusieurs couples de valeurs (S, P) . Dans cette expérience, c'est la mé-

| S | P | Meilleurs (γ, C) | 100-TMC |
|-----|-----|--|---------|
| 128 | 100 | $(5 \cdot 10^{-8}, 2.07 \cdot 10^3)$ | 86.3% |
| 128 | 200 | $(9.7 \cdot 10^{-7}, 4.7 \cdot 10^2)$ | 88.4% |
| 256 | 400 | $(4.8 \cdot 10^{-7}, 1.03 \cdot 10^3)$ | 91.3% |

TABLE 3 – Paramètres optimaux vs. base d'images

thode GWO, avec $T = 5$ itérations et $N_a = 25$ agents, qui

a estimé les paramètres optimaux (γ , C). Ceux-ci restent du même ordre malgré la variation de P . Quand P passe de 100 à 200, pour $S = 128$, les paramètres optimaux obtenus s'approchent de ceux obtenus avec la recherche fine ($S = 256$ et $P = 400$). De cela, et d'après les temps de calcul présentés en Table 2, on déduit qu'il est pertinent d'utiliser un algorithme en 3 phases, où les paramètres optimaux sont affinés itérativement en améliorant la base d'images.

4.3 Estimation des paramètres de classification optimaux : évaluation comparative

La Table 4 présente les résultats d'estimation de paramètres : PSO et GWO sont appliquées avec $T = 15$ itérations et $Na = 20$ agents, et avec $S = 256$ et $P = 400$; la méthode proposée aGWO est appliquée avec les paramètres donnés en section 3. Pour chaque méthode, le processus de classification est donc effectué 300 fois. On remarque que les valeurs de paramètres optimaux (γ , C) et les taux de classification obtenus par chaque méthode sont proches. En revanche, les temps de calcul obtenus par chaque méthode sont les suivants : 1970 secondes pour PSO ainsi que pour GWO, contre 968 secondes pour aGWO. Le temps de calcul est donc divisé par 2 comparé aux méthodes de l'état de l'art. Lorsque la valeur du paramètre η reste fixée à 1 au cours des trois étapes de aGWO, le temps de calcul est toujours de 968 sec. mais le taux de classification obtenu est de 91.7%.

| méthode | Meilleurs (γ , C) | 100-TMC |
|---------|---------------------------------------|---------|
| PSO | $(6.7 \cdot 10^{-7}, 6.0 \cdot 10^2)$ | 91.4% |
| GWO | $(1.6 \cdot 10^{-7}, 8.0 \cdot 10^2)$ | 91.5% |
| aGWO | $(4.9 \cdot 10^{-8}, 8.2 \cdot 10^1)$ | 91.9% |

TABLE 4 – Etude comparative

Nous avons également appliqué la méthode aGWO à un noyau linéaire. Elle a conduit à $C = 7.6 \cdot 10^1$ comme valeur optimale pour C et à 91.3% comme taux de bonne classification. Quand un noyau polynomial est utilisé, un degré $d = 2$ conduit à un taux de 91.5% avec des valeurs optimales de paramètres estimées par aGWO qui sont $\gamma = 118$, $C = 120$, and $c0 = 30$.

5 Conclusion

Pour la première fois, l'algorithme du GWO a été adapté pour la classification par genre d'images de visages. L'estimation des paramètres d'un classifieur de type SVM est effectuée en 3 phases : une version globale de GWO est appliquée sur une base réduite d'images pour estimer grossièrement ces paramètres ; puis cette estimation est ensuite affinée avec une version locale de GWO sur une base contenant des images plus nombreuses et plus grandes. La méthode de recherche des paramètres de classification est donc entièrement automatisée, et les

valeurs du paramètre η au cours des 3 phases de l'algorithme conviennent dans plusieurs cas. Sur la base FERET, la méthode proposée atteint un taux de bonne classification de 91.9% pour un temps de calcul divisé par deux par rapport aux méthodes de référence. Il semble pertinent, dans le futur, d'adapter des méthodes d'optimisation discrète pour diminuer encore le temps de calcul.

Références

- [1] B. Xia, B. B. Amor, D. Huang, M. Daoudi, Y. Wang, and H. Drira, "Enhancing gender classification by combining 3d and 2d face modalities," in *Signal processing conference (EUSIPCO), 2013 Proceedings of the 21st European*. IEEE, 2013, pp. 1–5.
- [2] L. Liu, P. Fieguth, G. Zhao, M. Pietikainen, and D. Hu, "Extended local binary patterns for face recognition," *Information Sciences*, vol. 358-359, pp. 56 – 72, 2016. [Online]. Available : //www.sciencedirect.com/science/article/pii/S0020025516302560
- [3] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, Perth, 1995, pp. 1942–1948.
- [4] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46 – 61, 2014. [Online]. Available : http://www.sciencedirect.com/science/article/pii/S0965997813001853
- [5] E. Elhariri, N. El-Bendary, A. E. Hassanien, and A. Abraham, "Grey wolf optimization for one-against-one multi-class support vector machines," in *2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, Nov 2015, pp. 7–12.
- [6] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [7] N. Mittal, U. Singh, and B. Singh Sohi, "Modified grey wolf optimizer for global engineering optimization," *Applied Computational Intelligence and Soft Computing*, vol. Article ID 7950348, p. 16 pages, 2016.
- [8] S. Mirjalili, "How effective is the grey wolf optimizer in training multi-layer perceptrons," *Applied Intelligence*, vol. 43, no. 1, pp. 150–161, 2015. [Online]. Available : http://dx.doi.org/10.1007/s10489-014-0645-7