Boosting k-NN classification for categorization of natural scenes

Paolo Piro[1,4]
Michel Barlaud[1] Richard Nock[2] Frank Nielsen[3]

[1]CNRS / University of Nice-Sophia Antipolis, Sophia Antipolis, France
[2]CEREGMIA / University of Antilles-Guyane
[3]Ecole Polytechnique, LIX, Palaiseau, France
[4]University Campus Bio-medico of Rome

Outline

Image categorization

Task

Automatically classify images into a discrete set of categories.

Challenges:

- large number of natural categories
- intra- / inter- class variability

Examples:

- global descriptors (GIST) + learning on training images [1]
- local keypoints descriptors (SIFT) + clustering (visual words) [2]

indoor

outdoor

tower

church

[1] *[A.Oliva, A.Torralba. IJCV 2001]*
[2] *[J. Sivic, A. Zisserman. ICCV 2003]*

k-NN classification

Uniform voting rule

Classify a query point using *majority vote* among the k closest training points.

Success:

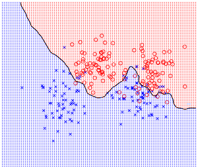
- naturally adapted to multi-class
- many possible prototypes per class
- very irregular decision boundary

Shortcomings:

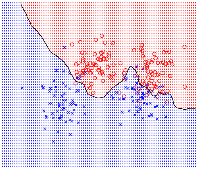
- significant bias in high dimensions [a]
- performance degradation when dealing with noisy examples
- high computational cost for large datasets

[a] *[Hastie,Tibshirani. PAMI 1996]*

k=11



k=17

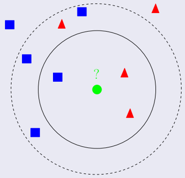


Leveraging of k-nearest neighbors (I)

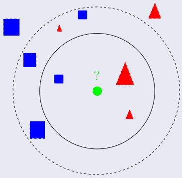
Notation:

- training example $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ (descriptor belonging to class $\tilde{c} \in \{1, 2, ..., C\}$)
- $\boldsymbol{y}_j \in \mathbb{R}^C$: "symmetric" class vector, $y_{jc} = \begin{cases} 1 & , \quad c = \tilde{c} \\ -\frac{1}{C-1} & , \quad c \neq \tilde{c} \end{cases}$
- $\boldsymbol{\alpha}_j \in \mathbb{R}^C$: vector of leveraging coefficients
- new observation $\boldsymbol{o}$ (image descriptor)
- $\mathrm{NN}_k(\boldsymbol{o})$: k-nearest neighbors examples

Leveraged k-NN rule



$$\boldsymbol{h}^{\ell}(\boldsymbol{o}) = \sum_{j \in \mathrm{NN}_k(\boldsymbol{o})} \boldsymbol{diag}(\alpha_j \boldsymbol{y}_j^{\top})$$



Leveraging of k-nearest neighbors (II)

Input: dataset of training examples $\mathcal{S} = \{\boldsymbol{x_i}, \boldsymbol{y_i},\ i = 1, 2, ..., m\}$

Definitions:

- weight distribution over the examples: $w_i,\ i = 1, 2, ..., m$
- k-NN "edge":

$$\mathrm{r}_{ij}^{(c)} = \begin{cases} y_{ic} y_{jc} & \text{if} \quad j \in \mathrm{NN}_k(\boldsymbol{o_i}) \\ 0 & \text{otherwise} \end{cases}$$

- w_j^+, w_j^-

$$\begin{array}{lll} \text{agreeing labels} & w_j^+ & = \sum_{i:\mathrm{r}_{ij}^{(c)}>0} w_i \\ \text{disagreeing labels} & w_j^- & = \sum_{i:\mathrm{r}_{ij}^{(c)}<0} w_i \end{array}$$

(sums are computed over the reciprocal nearest neighbors of j)

UNN: learning of leveraging coefficients α_{jc}

Algorithm 1: UNN (Universal Nearest Neighbors)

for $c = 1, 2, ..., C$ **do**

Let $\mathrm{r}_{ij}^{(c)} = \begin{cases} y_{ic} y_{jc} & \text{if} \quad j \in \mathrm{NN}_k(\boldsymbol{o}_i) \\ 0 & \text{otherwise} \end{cases}$

Let $\alpha_{jc} = 0$, $w_i = 1$, $\forall i, j = 1, 2, ..., m$;

for $t = 1, 2, ..., T$ **do**

[I.0] Let $j \leftarrow \textsc{Wic}(\{1, 2, ..., m\}, t)$;

[I.1] Let

$$w_j^+ = \sum_{i:\mathrm{r}_{ij}^{(c)} > 0} w_i,\ w_j^- = \sum_{i:\mathrm{r}_{ij}^{(c)} < 0} w_i\ , \tag{1}$$

$$\delta_j \leftarrow \frac{1}{2} \log \left(\frac{w_j^+}{w_j^-} \right); \tag{2}$$

[I.2] $\forall i$: $j \in \mathrm{NN}_k(\boldsymbol{o}_i)$, let

$$w_i \leftarrow w_i \exp(-\delta_j \mathrm{r}_{ij}^{(c)}); \tag{3}$$

[I.3] Let $\alpha_{jc} \leftarrow \alpha_{jc} + \delta_j$;

Properties of UNN

WIC (Weak Index Chooser oracle)

Oracle: selects index j of the next weak classifier
Implementations:

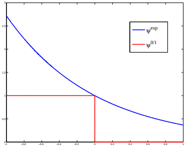
- *lazy*: select any weak classifier not yet used (either randomly or in alphabetic order)
- *boosting*: select the best weak classifier $j = \arg\max |\delta_j|$

Main properties:

1. minimization of the exponential risk
2. filtering of training data

Exponential risk minimization[3]

$$\varepsilon^{\psi}(\boldsymbol{h}, \mathcal{S}) = \frac{1}{mC} \sum_{c=1}^{C} \sum_{i=1}^{m} \psi(y_{ic} h_c(\boldsymbol{o_i})) \quad (4)$$



Parameterized wrt *surrogate loss* ψ:

$$\psi(x) = \exp(-x)$$

Upper bound to empirical risk: $\varepsilon^{0/1} \leq \varepsilon^{\psi}$, (empirical loss $0/1(x) = u(-x)$)

UNN convergence bound

If there exist $\gamma > 0$, $\eta > 0$ such that, for $\tau < T$:

- *weak learning assumption* $\frac{w_j^+ - w_j^-}{2(w_j^+ + w_j^-)} \geq \gamma > 0$
- *weak coverage assumption* $\frac{w_j^+ + w_j^-}{\sum_{i=1}^{m} w_i} \geq \eta > 0$

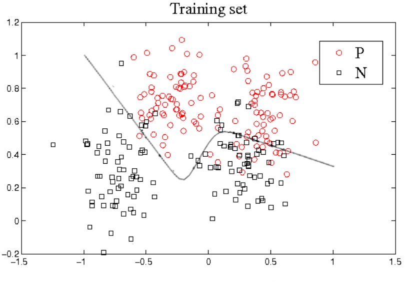
Then:

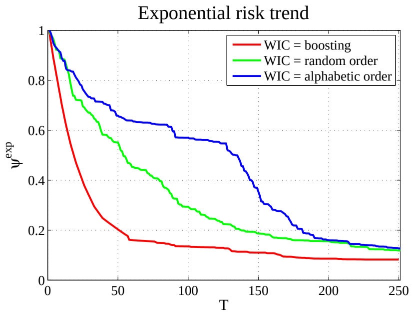
$$\varepsilon^{0/1}(\boldsymbol{h}^{\ell}, \mathcal{S}) \leq \varepsilon^{\psi}(\boldsymbol{h}^{\ell}, \mathcal{S}) \leq \exp(-2\eta\gamma^2\tau)$$

[3] *[R.Nock, F.Nielsen. On the Efficient Minimization of Classification-Calibrated Surrogates. NIPS 2008]*

Ripley's synthetic dataset

- 2 classes (P, N)
- each class is a mixture of two 2-d normal populations
- 250 training data, 1000 test data
- Bayes error of 8.0%



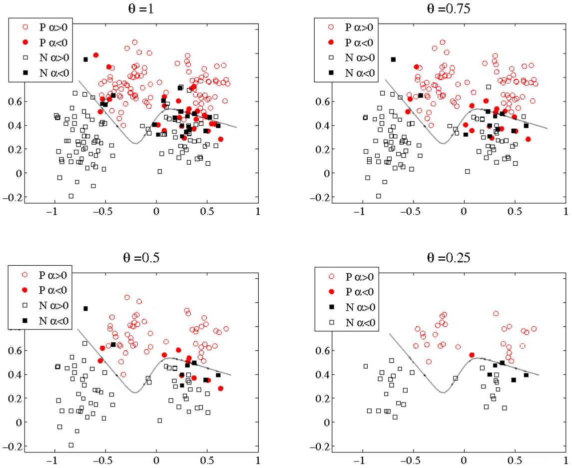


Filtering the training data

Leverage coefficients measure the *confidence* of training data

Filtering rule: retain a proportion θ of the examples with the largest α_j

Best UNN performance: **8.3% classification error** ($k = 9$, $\theta = 0.25$)



“Spatial envelope” database [4]

- 8 natural categories
- one GIST descriptor per image (dimension 512)
- 800 training images (100 per category), 1888 test images

coast

forest

highway

inside city

mountain

open country

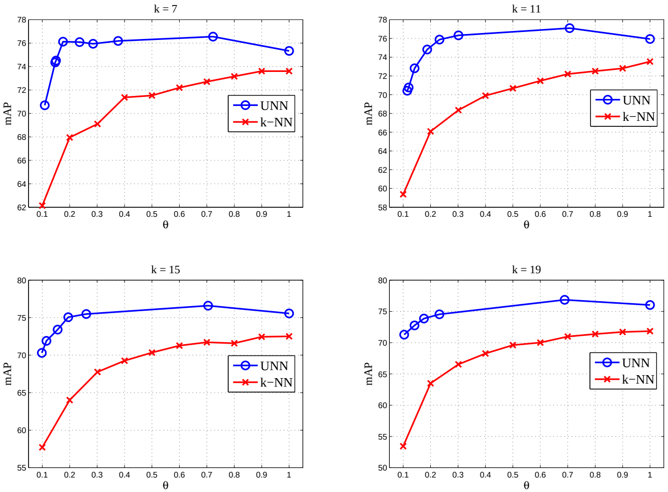
street

tall buildings

[4] *[A. Oliva, A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV, Vol. 42(3): 145-175, 2001.]*

Categorization performances (I)

Mean Average Precision of UNN compared to k-NN (random sampling)



Categorization performances (II)

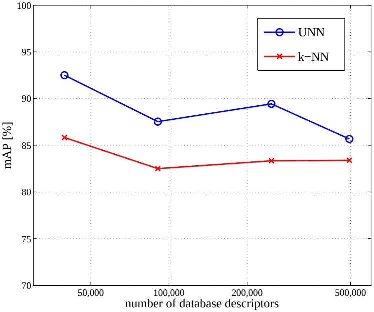
Confusion matrix [%]

Category	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1) Tall building	**77.3**	9.8	3.9	0.4	0.4	2.0	3.9	2.3
(2) Inside city	4.8	**69.7**	13.5	2.9	1.4	3.4	2.9	1.4
(3) Street	1.6	1.6	**89.1**	2.6	0.0	0.5	1.6	3.1
(4) Highway	0.0	2.5	2.5	**83.8**	5.6	3.1	2.5	0.0
(5) Coast	0.0	0.0	0.0	13.5	**76.5**	8.1	1.5	0.4
(6) Open country	0.3	0.6	1.6	7.4	14.5	**60.6**	11.0	3.9
(7) Mountain	0.7	1.5	2.2	3.6	3.3	9.5	**73.4**	5.8
(8) Forest	0.0	0.0	2.2	0.4	0.0	3.9	7.0	**86.4**

Mean Average Precision = **77.1%**

“Holidays” database [5]

- subset: 10 to 40 image groups
- about SIFT descriptors (dimension 128)
- max 69 training images ($\sim 500,000$ descriptors), 40 test images



[5] *[H. Jegou, M. Douze, C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. ECCV, 2008.]*

Conclusion

UNN for image categorization

- boosting of regular k-NN uniform voting (5% to 10% precision improvement)
- reducing the computational cost (up to an order of magnitude)

Perspectives:

- replace the Euclidean distance by other distorsion measures (e.g. Bregman k-NN retrieval [6])
- learning the distance metric for k-NN computation [7]

---

[6] *[F. Nielsen, P. Piro. Tailored Bregman Ball Trees for Effective Nearest Neighbors. EuroCG 2009]*

[7] *[K. Weinberger, J. Blitzer, L. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. NIPS 2005]*

Thanks!