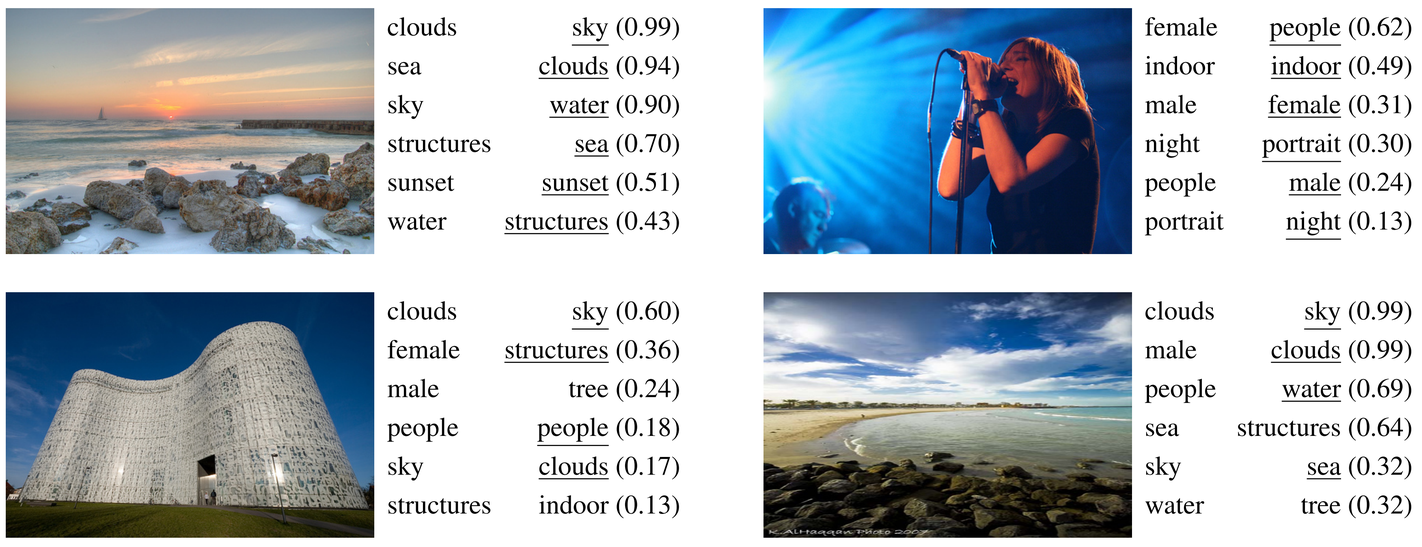
IMAGE ANNOTATION WITH TAGPROP ON THE MIRFLICKR SET

Verbeek, J., Guillaumin, M., Mensink, T., and Schmid, C.

LEAR team, INRIA Grenoble, Laboratoire Jean Kuntzmann

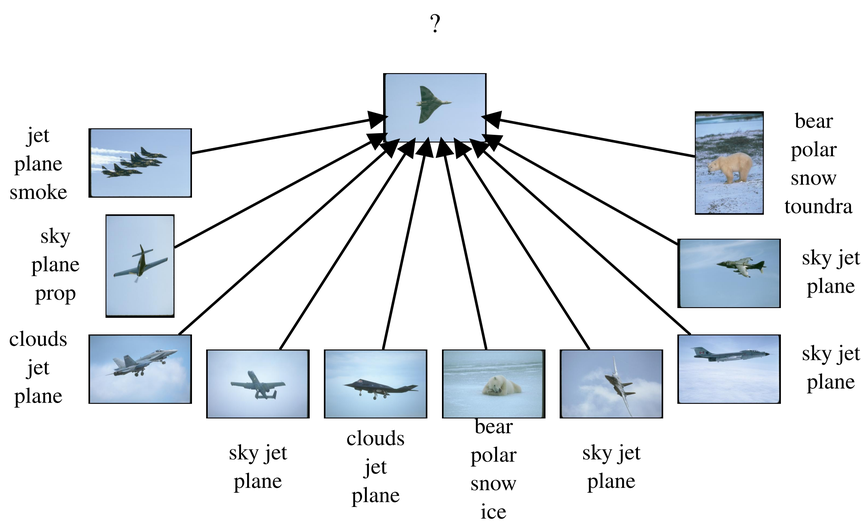
Summary

- Our goal is to predict keywords for images to support **image annotation**, and **keyword based retrieval**.
- The nearest neighbor approach is simple & effective, but: which distance, and how many neighbors?
- We present **TagProp**, a probabilistic nearest neighbor model, that learns these parameter from data.
- Different variants of TagProp are compared to SVMs learned for each annotation term.
- As additional features set, and to replace manual annotation we use Flickr tags.



EXAMPLES FROM THE DATA SET WITH ACTUAL ANNOTATION (LEFT), AND PREDICTED TERMS AND CONFIDENCE (RIGHT).

Weighted Nearest Neighbor Tag Prediction



- $y_{iw} \in \{0, 1\}$ denotes absence/presence of word w for image i
- π_{ij} is the weight for using image j for predicting tags of image i
- **Keyword presence probability is a weighted sum of keyword presence among visual neighbors** [1]:

$$p(y_{iw} = +1) = \textstyle\sum_j \pi_{ij} y_{iw} \quad (1)$$

- **Optimize leave-one-out log-likelihood** for tagprediction of training images
  - Set $\pi_{ii} = 0$ to avoid using image as neighbor of itself

$$\mathcal{L} = \sum_{i,w} c_{iw} \log p(y_{iw}), \quad (2)$$

  - c_{iw} is a cost to balance keyword presence/absence.
  - Often much more tag absences than presences, and absences are much noisier: *e.g.* photo sharing site, user gives a handful relevant tags.

Neighbor Weight Definitions

- **Rank-based**
  - Weight $\pi_{ij} = \gamma_k$ when j is k-th neighbor of i.
  - The effective neighborhood size is given by the γ_k.
  - Combine several image distances by $\pi_{ij} = \sum_d \pi_{ij}^d$.
- **Distance-based**
  - Weight it a smooth function of image distances:

$$\pi_{ij} = \frac{\exp\left(-\lambda d_{ij}\right)}{\sum_k \exp\left(-\lambda d_{ik}\right)}, \quad (3)$$

  - Single parameter controls decrease of weights with distance.
  - Combine distances with linear combinations:

$$d_{ij} = \mathbf{w}^\top \mathbf{d}_{ij}$$

.

[1] Guillaumin M., Mensink T., Verbeek J., Schmid C. TagProp: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, 2009.

Word-Specific Logistic Discriminant

- **Correct for different frequency of words in database**
  - Frequent tags appear too often in predicted annotations
- Boost the probability of rare keywords, and suppress it for frequent ones, using word-specific logistic discriminant model

$$p(y_{iw} = +1) = \sigma(\alpha_w x_{iw} + \beta_w), \text{with } x_{iw} = \sum_j \pi_{ij} y_{jw} \quad (4)$$

- Does not change the image ordering for a specific keyword, but only the keyword ordering for a specific image.

Training TagProp

- Iterative optimization with fast convergence (~ 3 iterations).
  - Update π_{ij}, using projected gradient to enforce constraints.
  - Update $\{\alpha_w, \beta_w\}$ for all words, concave for fixed π_{ij}
- To obtain linear scaling with # images: compute π_{ij} only over K nearest neighbors, others are assumed to be zero.

Image Representations

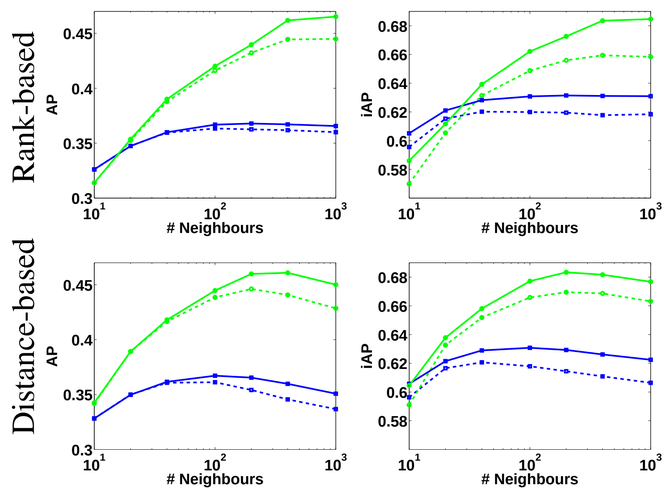
- **Combination of 15 local and global feature sets**
  - Global: GIST, and color histograms (RGB,HSV,LAB)
  - Local: SIFT and robust Hue histograms
- Local descriptors computed on grid and interest points
- Image layout captured by features from spatial 3×1 grid.

Performance Evaluation

- **Mean Average Precision** over keywords (AP, retrieval)
  - How well do we rank images for keywords
- Mean Average Precision over images (iAP, annotation)
  - How well do we rank keywords for images

Evaluating TagProp variants

- Rank-based or distance-based weights
- With sigmoid (solid) or without (dashed)
- Fixed distance combination (blue) or learned (green)



Flickr Tags as Additional Features

- **Binary features 457 tags with at least 50 occurrences**
- Support Vector Machines trained on true image labels
- We compare different feature sets, and combinations
  - **V**isual features: using RBF kernel on image distances
  - **T**ag features: binary vector indicates tag presence
  - **P**redicted tags: vector with TagProp predictions
- Kernel averaging to combine features, equivalent to concatenating corresponding feature vectors.

	TagProp Dist	TagProp Rank	SVM-V	SVM-T	SVM-P	SVM V+P	SVM T+P	SVM V+T	SVM V+T+P
mean AP	45.9	46.5	52.4	43.7	43.6	53.0	58.8	63.3	64.0

MEAN AP OVER ALL CONCEPTS

1. **Combination T+P performs surprisingly well at 58.8, much better than using visual features alone (V) at 52.4**
2. Visual-only SVM outperforms TagProp, but learning SVMs per concept is slow (hours instead of seconds)
3. Tag features (T) and TagProp predictions (P) perform similar
4. TagProp predictions (P) add little if visual features (V) are already included: due to dependency among feature sets.

Flickr Tags as Noisy Image Labels

- **Using the 18 concepts names that also appear as tag**
- TagProp and SVMs trained on tag absence/presence
  - Kernel averaging to combine features for SVM as before

	TagProp Dist	TagProp Rank	SVM-V	SVM-T	SVM-P	SVM V+P	SVM T+P	SVM V+T	SVM V+T+P
AP	38.4	37.4	35.4	23.0	24.9	35.0	31.7	37.9	**38.7**
iAP	**47.3**	46.3	44.2	32.0	36.4	44.5	42.5	45.0	46.2

MEAN AP OVER THE 18 CONCEPTS

1. **TagProp more resistant than SVMs to training label noise**
2. Including user tags SVMs perform similar to TagProp
3. Worse performance from tags than from manual labels