TagProp: Discriminative Metric Learning in Nearest Neighbour Models for Image Auto-Annotation

Guillaumin, Mensink, Verbeek, Schmid

Daniel Rios-Pavia, Thomas Vincent-Sweet

UJF, Ensimag

January 14, 2011

Layout

Layout

TagProp: Tag Propagation

- **Aim:** Tag images automatically through keyword relevance prediction
- **Applications:**
  - Image annotation
  - Image search

Auto-Annotation Example

	true	predicted
	box	box (1.00)
	brown	square (1.00)
	square	brown (1.00)
	white	white (0.79)
		yellow (0.72)

	true	predicted
	glacier	glacier (1.00)
	mountain	mountain (1.00)
	people	front (0.64)
	tourist	sky (0.58)
		people (0.58)

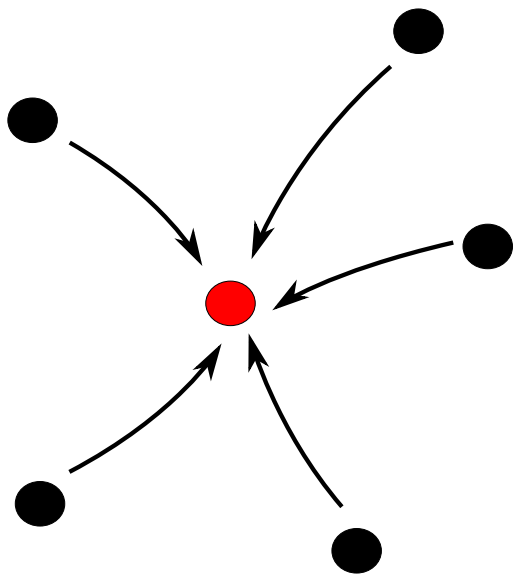
	true	predicted
	blue	man (0.98)
	cartoon	anime (0.96)
	man	cartoon (0.92)
	woman	people (0.89)
		woman (0.88)

	true	predicted
	landscape	llama (1.00)
	lot	water (1.00)
	meadow	landscape (1.00)
	water	front (0.60)
		people (0.51)

Layout

Predicting Tag Relevance

- Propagate annotations from training images to new images
- Use metric learning instead of fixed metric or ad-hoc combinations of metrics



Weighted Nearest Neighbour Tag Prediction

- Tags are either absent or present (i: image w: word)

$$y_{iw} \in \{-1, +1\}$$

Weighted Nearest Neighbour Tag Prediction

- Tags are either absent or present (i: image w: word)

$$y_{iw} \in \{-1, +1\}$$

- Tag presence prediction $p(y_{iw} = +1)$:

$$p(y_{iw} = +1) = \sum_j \pi_{ij}\ p(y_{iw} = +1|j)$$

$$p(y_{iw} = +1|j) = \begin{cases} 1 - \epsilon & \text{for } y_{jw} = +1 \\ \epsilon & \text{otherwise} \end{cases}$$

with π_{ij} the weight of training image j for predictions for image i.

Weighted Nearest Neighbour Tag Prediction

- Tags are either absent or present (i: image w: word)

$$y_{iw} \in \{-1, +1\}$$

- Tag presence prediction $p(y_{iw} = +1)$:

$$p(y_{iw} = +1) = \sum_j \pi_{ij}\ p(y_{iw} = +1|j)$$

$$p(y_{iw} = +1|j) = \begin{cases} 1 - \epsilon & \text{for } y_{jw} = +1 \\ \epsilon & \text{otherwise} \end{cases}$$

with π_{ij} the weight of training image j for predictions for image i.

- $\pi_{ij} \geq 0$ and $\sum_j \pi_{ij} = 1$

Weighted Nearest Neighbour Tag Prediction

- Estimation of parameters that control weights π_{iw}
  - Maximize the log-likelihood of predictions:

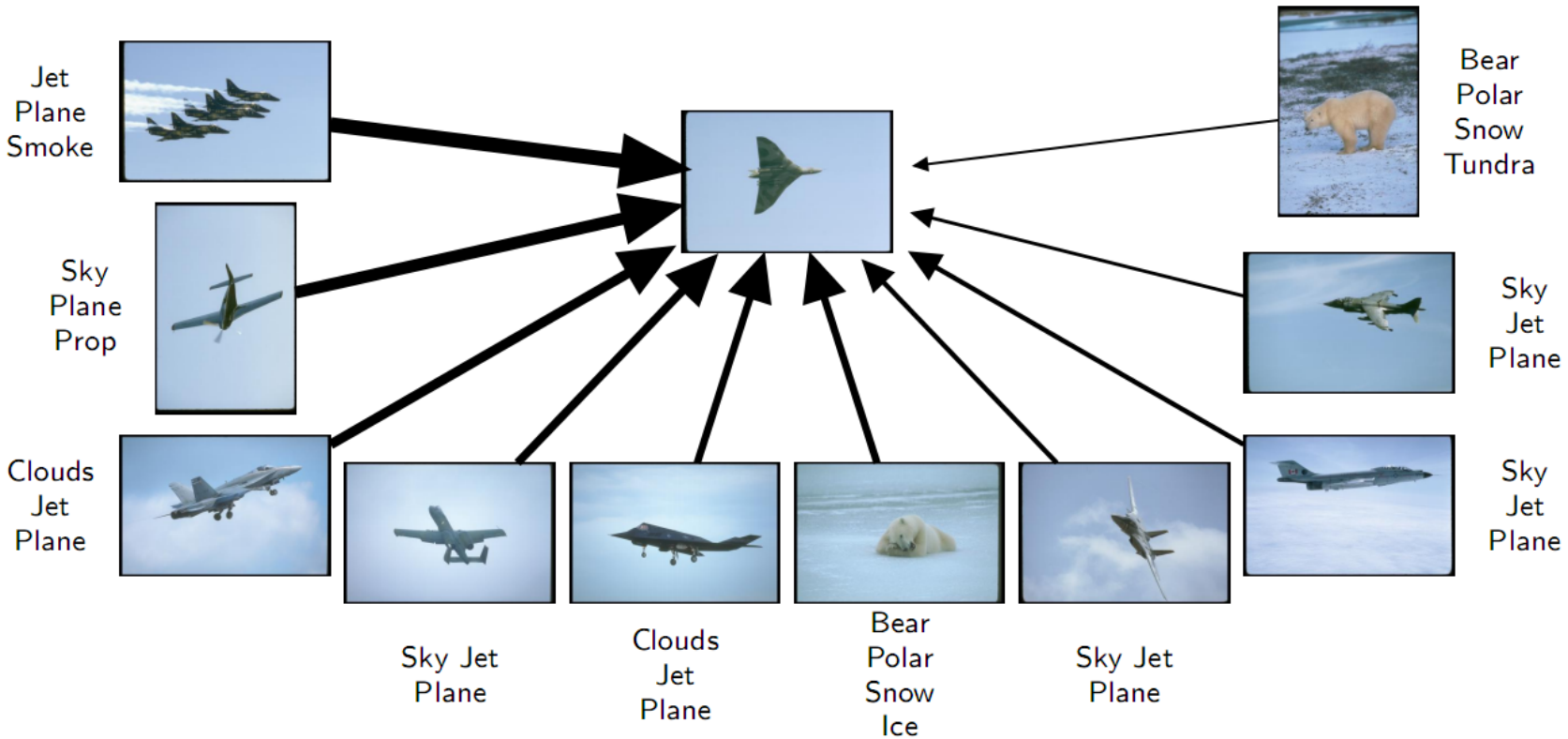
$$\mathcal{L} = \sum_{i,w} c_{iw} \log p(y_{iw})$$

where c_{iw} is the cost taking into account presence/absence imbalance:

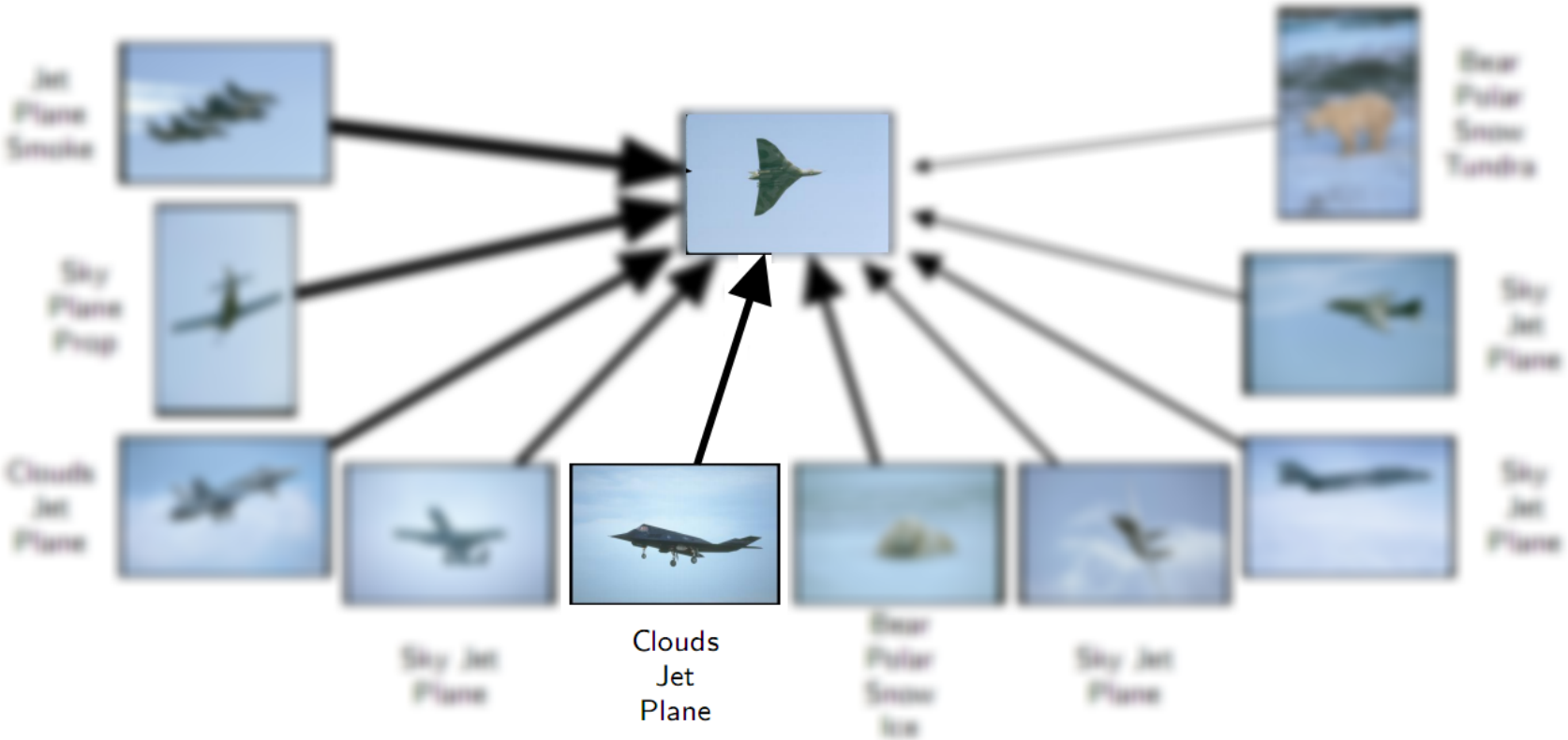
$$c_{iw} = \frac{1}{n^+} \text{ if } y_{iw} = +1$$

n^+ being the total number of positive labels. (same for n^-)

Example


Jet
Plane
Smoke
Sky
Plane
Prop
Clouds
Jet
Plane
Sky Jet
Plane
Clouds
Jet
Plane
Bear
Polar
Snow
Ice
Sky Jet
Plane
Bear
Polar
Snow
Tundra
Sky
Jet
Plane
Sky
Jet
Plane

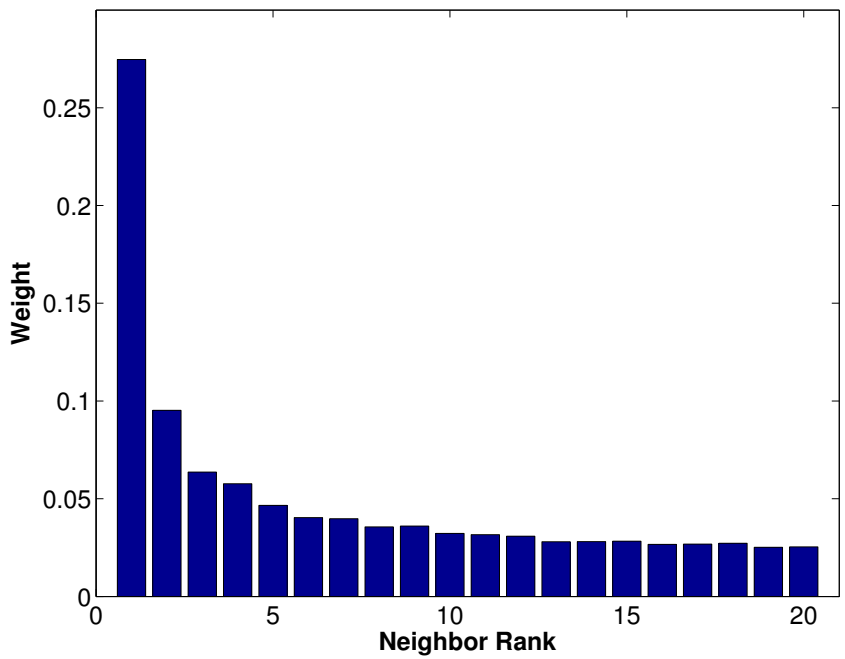
Example



$$p(y_{iw} = +1|j) = \begin{cases} 1-\epsilon & \text{for } y_{jw} = +1 \\ \epsilon & \text{otherwise} \end{cases}$$

Rank-based Weighting

- Fixed weight for the k-th neighbor: $\pi_{iw} = \gamma_k$
- K neighbors $\longrightarrow$ K parameters
- $\mathcal{L}$ is concave with respect to $\{\gamma_k\}$
  - EM-algorithm
  - Projected gradient descent
- Effective neighborhood size is set automatically



Distance-based Weighting

- Weights given by visual distance d_θ

$$\pi_{ij} = \frac{\exp(-d_\theta(i,j))}{\sum_k \exp(-d_\theta(i,k))}$$

  where θ are the parameters we want to optimize.

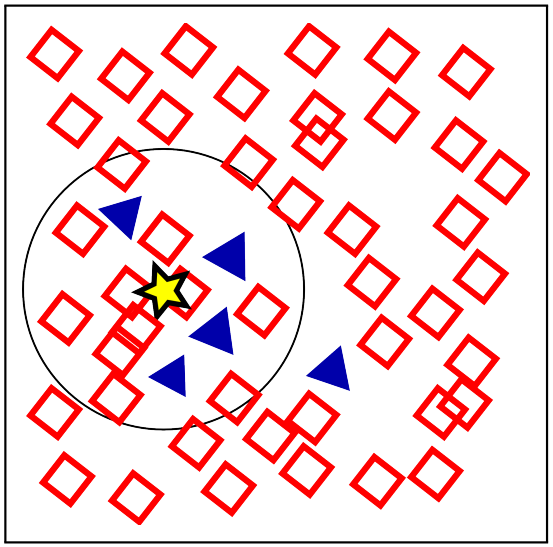
- Weights depend smoothly on distance
  - important if distance adjustment is needed during training.
- Only one parameter per base distance

Distance-based Weighting

- Choices for d_θ include (not exhaustive):
  - A fixed distance d with a positive scale factor
  - $d_w(i,j) = w^T d_{ij}$ with d_{ij} a vector of base distances w contains the positive coefficients of the distance combination
  - Mahalanobis distance
- As before, projected gradient algorithm to maximize log-likelihood and learn the distance combination.

Boosting the Recall of Rare Words

- Keywords with low frequency in database have low recall
  - Mass of neighbors too small
  - Systematic low relevance of keyword
- Boosting needed.



Sigmoidal modulation

- Word-specific logistic discriminant model
  - 'dynamic range' adjusted per word

$$p(y_{iw} = +1) = \sigma(\alpha_w x_{iw} + \beta_w)$$

Sigmoidal modulation

- Word-specific logistic discriminant model
  - 'dynamic range' adjusted per word

$$p(y_{iw} = +1) = \sigma(\alpha_w x_{iw} + \beta_w)$$

$$\text{with } \sigma(z) = \frac{1}{(1 + \exp(-z))}$$

$$\text{and } x_{iw} = \sum_j \pi_{ij}\ y_{iw}$$

Sigmoidal modulation

- Word-specific logistic discriminant model
  - 'dynamic range' adjusted per word

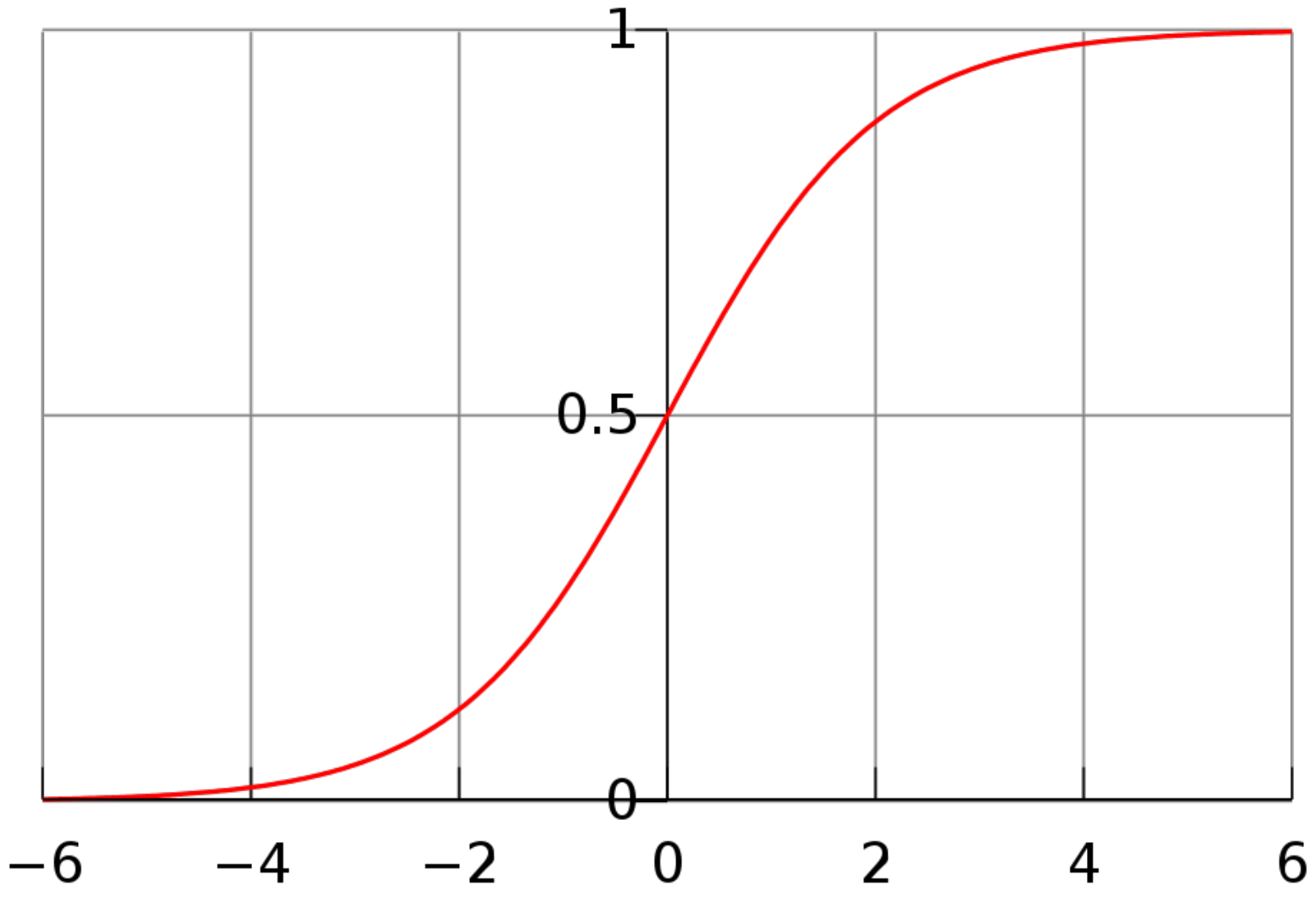
$$p(y_{iw} = +1) = \sigma(\alpha_w x_{iw} + \beta_w)$$

$$\text{with } \sigma(z) = \frac{1}{(1 + \exp(-z))}$$

$$\text{and } x_{iw} = \sum_j \pi_{ij}\ y_{iw}$$

- Adds 2 parameters for each word $\{\alpha_w, \beta_w\}$
- Optimize through training (alternating maximization)
  - $\{\alpha_w, \beta_w\}$
  - neighbour weights π_{ij}

Sigmoid function



Layout

Feature Extraction

- **15 image representations:**

Feature Extraction

- **15 image representations:**
- **Global GIST descriptor**
- **Global colour histograms**
  - RGB, HSV, LAB
  - 16 bin quantization
- **Bag-of-Words histograms**
  - SIFT and Hue descriptors
  - Dense grid and Harris-Laplacian interest points
  - K-means quantization

Feature Extraction

- **15 image representations:**
- **Global GIST descriptor**
- **Global colour histograms**
  - RGB, HSV, LAB
  - 16 bin quantization
- **Bag-of-Words histograms**
  - SIFT and Hue descriptors
  - Dense grid and Harris-Laplacian interest points
  - K-means quantization
- **3x1 spatial partitioning** for BoW and colour histograms

Corel 5k

- 5000 images (landscape, animals...)
- max 5 tags per image (avg=3)
- Vocabulary size = 260

ESP Game

- 20'000 images subset - 60k total (drawings, photos...)
- max 15 tags per image (avg=5)
- Vocabulary size = 268
- Players annotate images in pairs

IAPR TC12

- 20'000 images (tourist photos, sports...)
- max 23 tags per image (avg=6)
- Vocabulary size = 291
- Natural language processing from descriptive text

Evaluation Method

- Compute measures per keyword, then average
- Annotate images with top 5 keywords
  - **Recall** (nr. annotated/nr. in DB)
  - **Precision** (nr. correctly annotated/nr.annotated)
  - **N+** (nr. words with recall > 0)
- Retrieval (search)
  - Rank results according to query keyword presence probability
  - Precision for n_w images (nr. ground truth images with w)
  - Mean Average Precision (**mAP**) and Break-Even Point (**BEP**)

Layout

Results: Annotation

	COREL 5K		IAPR TC-12		ESP Game	
	P	R	P	R	P	R
Rank-based	28%	32%	35%	22%	27%	20%
Fixed distance	30%	33%	**50%**	20%	**48%**	19%
Fixed with sigmoid	28%	35%	41%	30%	39%	24%
ML with sigmoid	**33%**	**42%**	46%	**35%**	39%	**27%**

- Distance $>$ Rank

Results: Annotation

	COREL 5K		IAPR TC-12		ESP Game	
	P	R	P	R	P	R
Rank-based	28%	32%	35%	22%	27%	20%
Fixed distance	30%	33%	**50%**	20%	**48%**	19%
Fixed with sigmoid	28%	35%	41%	30%	39%	24%
ML with sigmoid	**33%**	**42%**	46%	**35%**	39%	**27%**

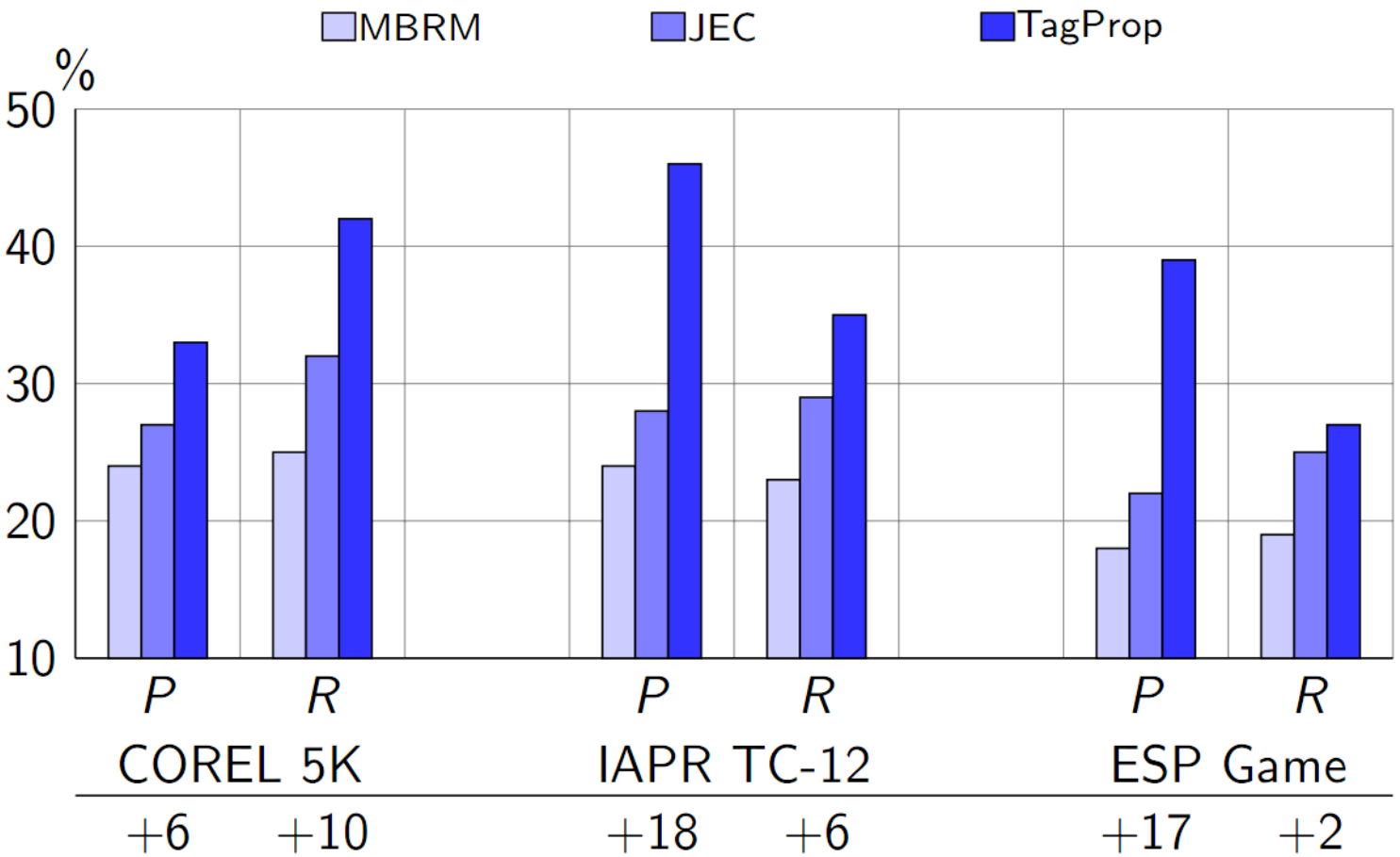
- Distance $>$ Rank
- Sigmoid improves recall, loses precision

Results: Annotation

	COREL 5K		IAPR TC-12		ESP Game	
	P	R	P	R	P	R
Rank-based	28%	32%	35%	22%	27%	20%
Fixed distance	30%	33%	**50%**	20%	**48%**	19%
Fixed with sigmoid	28%	35%	41%	30%	39%	24%
ML with sigmoid	**33%**	**42%**	46%	**35%**	39%	**27%**

- Distance $>$ Rank
- Sigmoid improves recall, loses precision
- Metric Learning gives significantly better results!

Results Improvement



Results: Recall

	All	Single	Multi	Easy	Difficult	All-**BEP**
PAMIR [7]	26	34	26	43	22	17
WN	32	40	31	49	28	24
σWN	31	41	30	49	27	23
WN-ML	**36**	43	**35**	53	**32**	**27**
σWN-ML	**36**	**46**	**35**	**55**	**32**	**27**

Annotation example: Corel 5k

BEP: *100%*
Ground Truth: **sun** *(1.00)*, **sky** *(1.00)*, **tree** *(1.00)*, **clouds** *(0.99)*
Predictions: **sun** *(1.00)*, **sky** *(1.00)*, **tree** *(1.00)*, **clouds** *(0.99)*

BEP: *100%*
Ground Truth: **mosque** *(1.00)*, **temple** *(1.00)*, **stone** *(1.00)*, **pillar** *(1.00)*
Predictions: **mosque** *(1.00)*, **temple** *(1.00)*, **stone** *(1.00)*, **pillar** *(1.00)*

BEP: *50%*
Ground Truth: **grass** *(0.98)*, **tree** *(0.98)*, bush *(0.54)*, truck *(0.05)*
Predictions: flowers *(1.00)*, **grass** *(0.98)*, **tree** *(0.98)*, moose *(0.95)*

BEP: *50%*
Ground Truth: **herd** *(0.99)*, **grass** *(0.98)*, tundra *(0.96)*, caribou *(0.13)*
Predictions: sky *(0.99)*, **herd** *(0.99)*, **grass** *(0.98)*, hills *(0.97)*

BEP: *50%*
Ground Truth: **mountain** *(1.00)*, **tree** *(0.99)*, sky *(0.98)*, clouds *(0.94)*
Predictions: hillside *(1.00)*, **mountain** *(1.00)*, valley *(0.99)*, **tree** *(0.99)*

Retrieval example: Corel 5k

tiger *100.00 (10)*

garden *60.00 (10)*

town *22.22 (9)*

water, pool *90.00 (10)*

beach, sand *25.00 (8)*

Layout

Conclusion

- State-of-the-art results!
- Contributions
  - Metric learning (no manual tuning)
  - Sigmoidal boosting for rare word recall