Pair-wise Ranking in visual domain

Learning to Rank

- Point-wise ranking
- **Pair-wise ranking**
- List-wise ranking

Disadvantage of Point-wise ranking

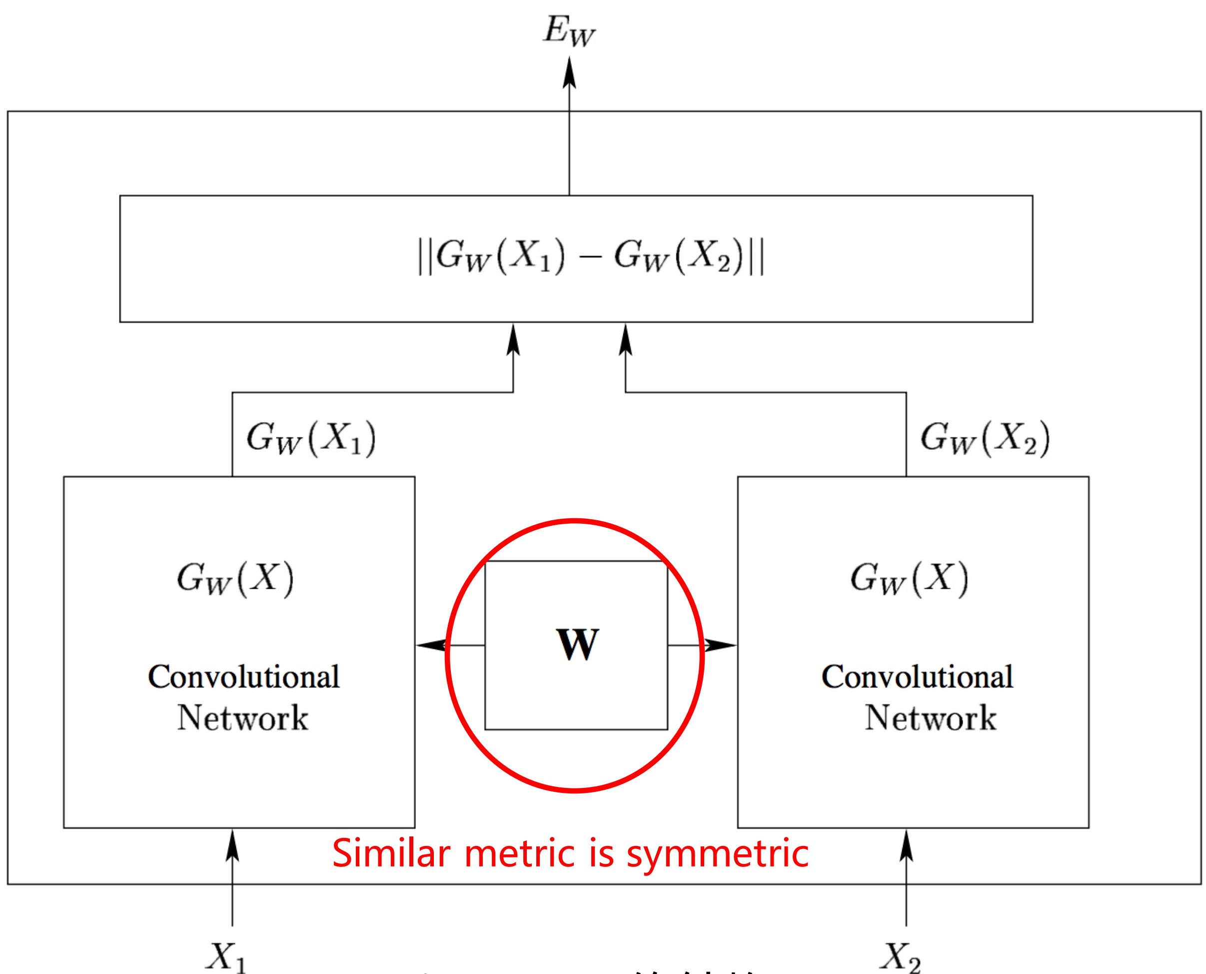
Pair-wise Ranking

- **Siamese Network/ Triplet Network**
- **Rank SVM**
- **RankNet**
- RankBoost

Opensource: http://people.cs.umass.edu/~vdang/ranklib.html

Siamese/Triplet Network

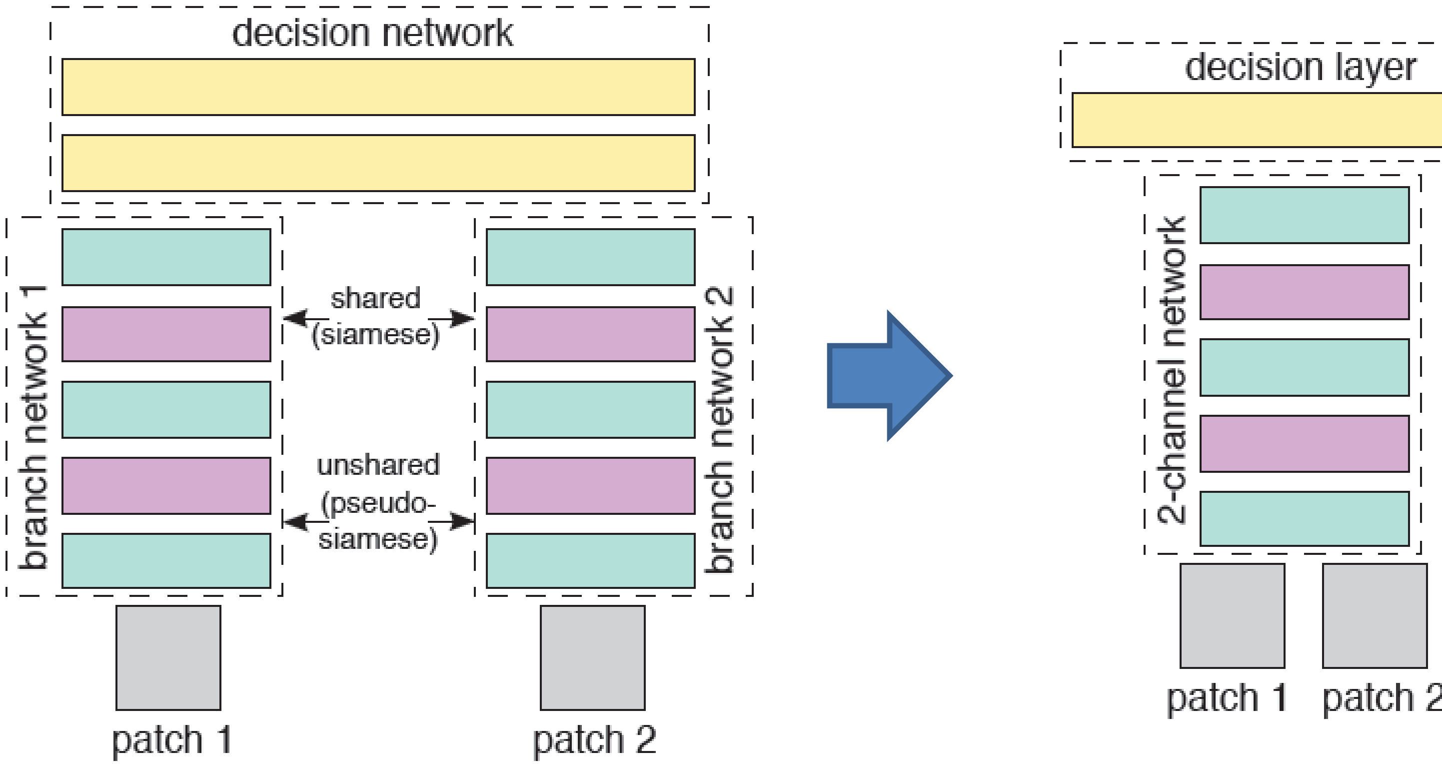
Siamese Network



Siamese网络结构

Contrastive Loss: $L(W, Y, X_1, X_2) = (1 - Y) L_G(E_W) + Y L_I(E_W)$

Siamese Network

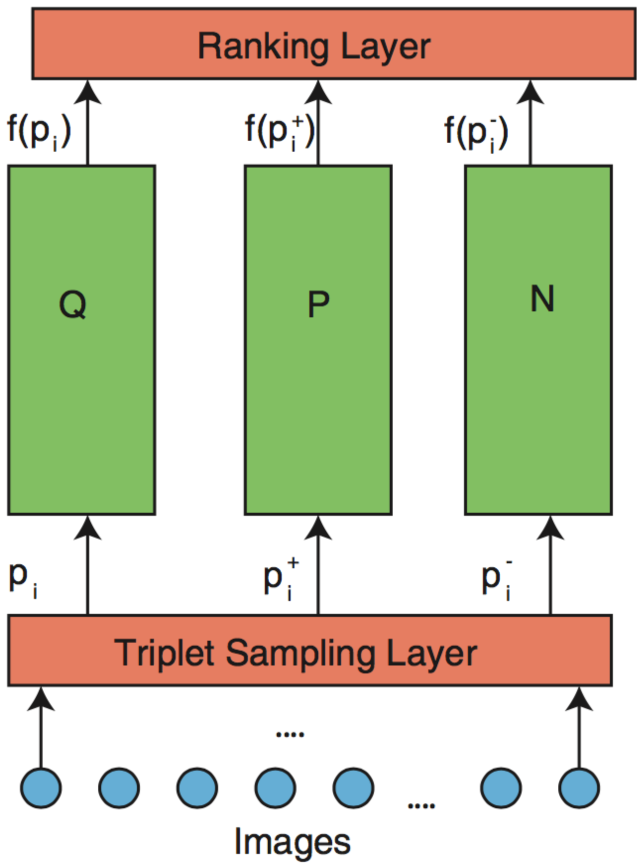


Siamese/Pseudo-siamese

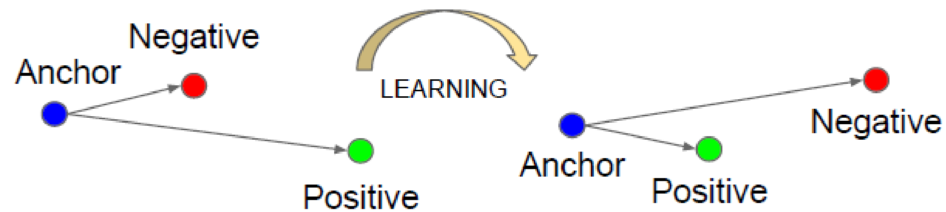
2-channel net

Sergey Zagoruyko, Nikos Komodakis. Learning to Compare Image Patches via Convolutional Neural Networks. CVPR 2015.

Triplet Network



Model structure



Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, Ying Wu. Learning Fine-grained Image Similarity with Deep Ranking. CVPR 2014.

Triplet Network

Pair-wise relevance score: $r_{i,j} = r(p_i, p_j)$

Total relevance score of image i: $r_i = \sum_{j:c_j=c_i, j\neq i} r_{i,j}$

Probability of choosing positive: $P(p_i^+) = \frac{\min\{T_p, r_{i,i^+}\}}{Z_i}$

Select in-class negative: $r_{i,i^+} - r_{i,i^-} \geq T_r, \forall t_i = (p_i, p_i^+, p_i^-)$

Triplet Sampling Strategy

Triplet Network

Hinge loss :

$$loss = \max(0, margin - (d(q,t^+) - d(q,t^-)))$$

其中 $d(,)$ 是距离（相似度）度量函数

- ℓ_2-norm : $d(x,y) = ||x-y||_2^2 = \sum_{i=1}^{n}(x_i - y_i)^2$
- 余弦相似度 : $d(x,y) = \frac{x \cdot y}{||x|| \cdot ||y||}$

Application of Siamese/Triplet network

➢ Face vertification or Face recognition

1. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. CVPR 2005.
2. FaceNet. CVPR 2015

➢ Fine-grained image similarity

1. Sergey Zagoruyko, Nikos Komodakis . Learning to Compare Image Patches via Convolutional Neural Networks. CVPR 2015.
2. Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, Ying Wu. Learning Fine-grained Image Similarity with Deep Ranking. CVPR 2014.

➢ Image retrieval

1. Junshi Huang, Rogerio Feris, Qiang Chen, Shuicheng Yan. Cross-domain Image Retrieval with a Dual Attribute-aware Ranking Network. ICCV 2015.

Opensource:
Siamese Network: http://caffe.berkeleyvision.org/gathered/examples/siamese.html
Triplet Netword: https://github.com/xiaolonw/caffe-video_triplet

Rank-SVM

Rank SVM

采用Kendall's τ来统计实际排序与算法排序的度量： r_a为真实排序，r_b为算法排序

1. P表示排序序列中保持一致性的`Pair`对数量，也就是真实相关性高的排在第的前面。
2. Q表示排序序列中保持不一致的`Pair`对数量（就是为逆序了），也就是由于算法的误差导致真实相关性低的排在了高的前面
3. 同时$P+Q=\binom{m}{2}$，m表示序列中文档的数量，因为长度为m的序列可能组成的`pair`对为m的2组合

则τ的计算方式为:

$$\tau(r_a, r_b) = \frac{P-Q}{P+Q} = 1 - \frac{2Q}{\binom{m}{2}}$$

假设现在有n个q_i作为训练样本，他们各自的目标排序为r_i^*，也就是:

$$(q_1, r_1^*), (q_2, r_2^*), (q_3, r_3^*), \ldots (q_n, r_n^*)$$

其中算法排序为$\hat{r}_i$，则排序算法的优化目标是将下列式子

$$\tau_s = \frac{1}{n}\sum_{i=1}^{n}\tau(r_i^*, \quad \hat{t}_i)$$

进行最大化.

Rank SVM

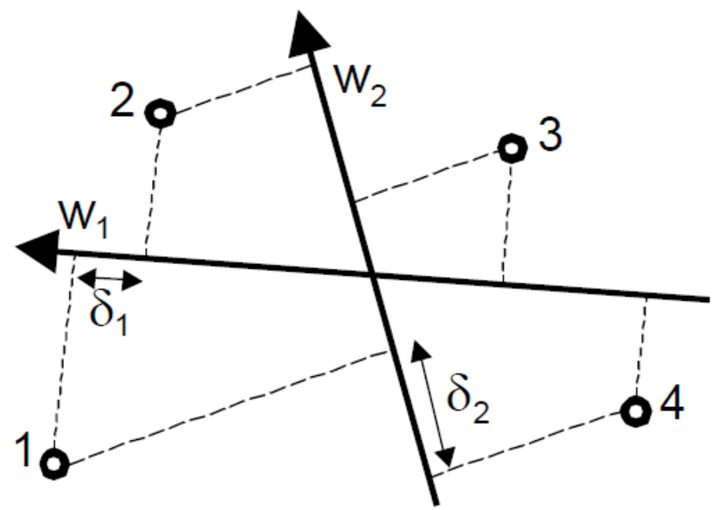
$$(d_i, d_j) \in \hat{r} \Leftrightarrow \vec{w}\Phi(q, d_i) > \vec{w}\Phi(q, d_j)$$

$$\forall(\mathrm{d}_i, \mathrm{d}_j) \in \mathrm{r}_1^* : \quad \vec{w}\Phi(\mathrm{q}_1, \mathrm{d}_i) > \vec{w}\Phi(\mathrm{q}_1, \mathrm{d}_j)$$
$$\ldots$$
$$\forall(\mathrm{d}_i, \mathrm{d}_j) \in \mathrm{r}_n^* : \quad \vec{w}\Phi(\mathrm{q}_n, \mathrm{d}_i) > \vec{w}\Phi(\mathrm{q}_n, \mathrm{d}_j)$$

minimize: $$V(\vec{w}, \vec{\xi}) = \frac{1}{2}\vec{w} \cdot \vec{w} + C \sum \xi_{i,j,k}$$

subject to:

$$\forall(d_i, d_j) \in r_1^* : \vec{w}\Phi(q_1, d_i) \geq \vec{w}\Phi(q_1, d_j) + 1 - \xi_{i,j,1}$$
$$\ldots$$
$$\forall(d_i, d_j) \in r_n^* : \vec{w}\Phi(q_n, d_i) \geq \vec{w}\Phi(q_n, d_j) + 1 - \xi_{i,j,n}$$
$$\forall i \forall j \forall k : \xi_{i,j,k} \geq 0$$



Thorsten Joachims. Optimizing Search Engines using Clickthrough Data. SIGKDD 2002.

Rank SVM

$$\vec{w}\Big(\Phi(q,d_i)-\Phi(q,d_j)\Big)\geq 1-\xi_{i,j,1}$$

定义di排在dj前面的Pair为正标签，否则为负标签:

$$y=\begin{cases}+1 & if \quad \vec{w}\Phi(q,d_i)>\vec{w}\Phi(q,d_j)\\ -1 & otherwise\end{cases}$$

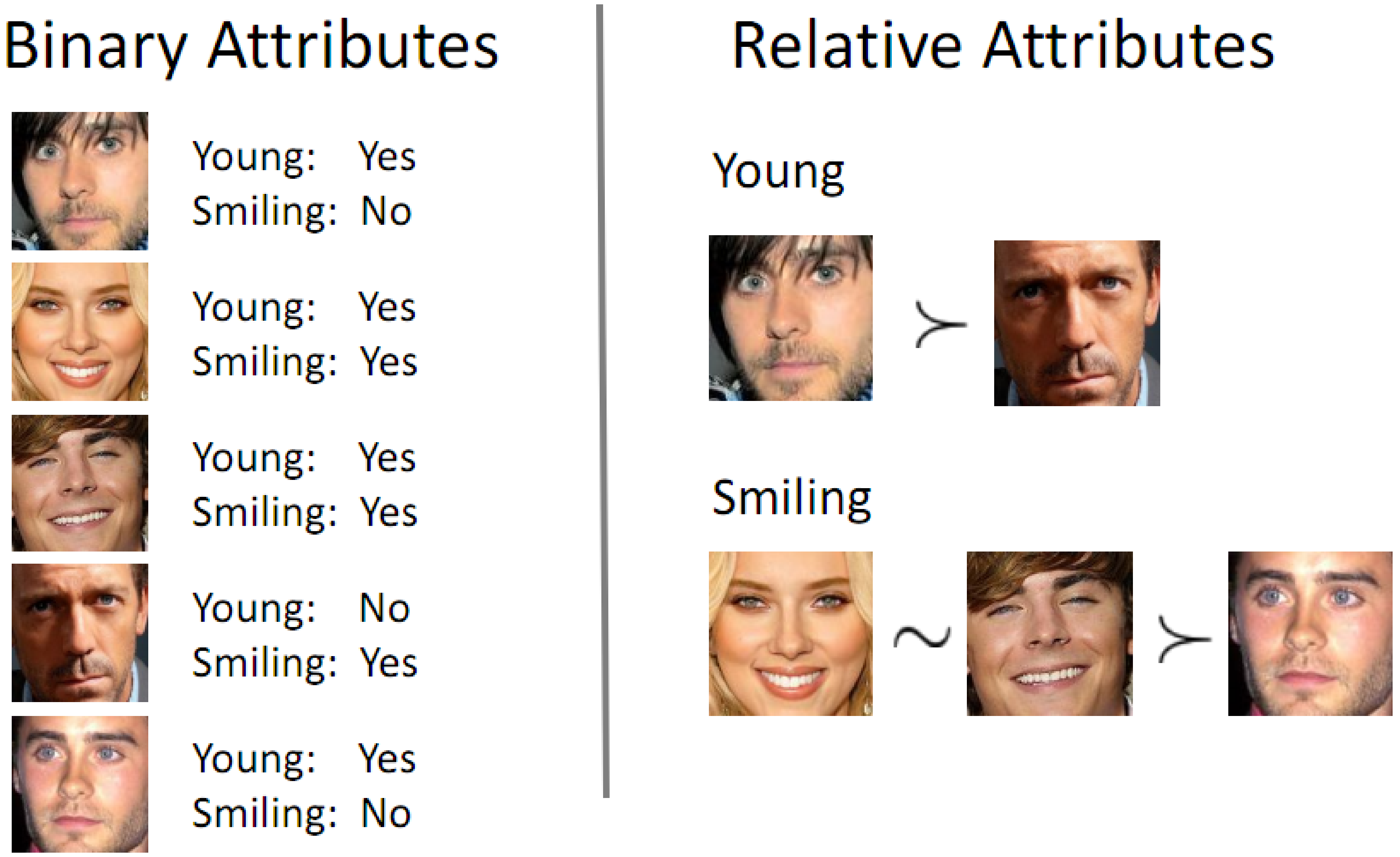
$$y_i\cdot\vec{w}\Big(\Phi(q,d_i)-\Phi(q,d_j)\Big)\geq 1-\xi_{i,j,1}$$ ⇨ 分类问题，可用SVM的对偶形式进行求解

排序时只需要将原始特征向量输入SVM模型即可

$$resv(q,d_i)=\vec{w}\Phi(q,d_i)=\sum_{l}^{n}a_l^{*}y_l(\Phi(q,d_i)\cdot\Phi(q,d_l))$$

Thorsten Joachims. Optimizing Search Engines using Clickthrough Data. SIGKDD 2002.

Applications

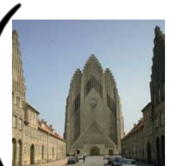
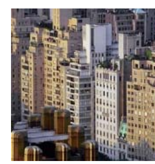


Relative attribute indicates **<u>the strength of an attribute</u>** in an image **<u>with respect to other image</u>** rather than simply predicting the presence of an attribute.

Learning Relative Attributes

For each attribute a_m, open

Supervision is

$O_m : \Big\{\Big($  $\succ$  $\Big), \dots \Big\},$

$S_m : \Big\{\Big\{$  $\sim$  $\Big\}, \dots \Big\}$

Learning Relative Attributes

Image features

Learn a scoring function $r_m(\boldsymbol{x_i}) = \boldsymbol{w_m^T x_i}$

Learned parameters

that best satisfies constraints:

$$\forall (i,j) \in O_m : \boldsymbol{w_m^T x_i} > \boldsymbol{w_m^T x_j}$$

$$\forall (i,j) \in S_m : \boldsymbol{w_m^T x_i} = \boldsymbol{w_m^T x_j}$$

Learning Relative Attributes

Max-margin learning to rank formulation

$$\min \quad \left(\frac{1}{2}||\boldsymbol{w_m^T}||_2^2 + C\left(\sum \xi_{ij}^2 + \sum \gamma_{ij}^2\right)\right)$$

$$\text{s.t} \quad \boldsymbol{w_m^T}(\boldsymbol{x_i} - \boldsymbol{x_j}) \geq 1 - \xi_{ij}, \forall (i,j) \in O_m$$

$$|\boldsymbol{w_m^T}(\boldsymbol{x_i} - \boldsymbol{x_j})| \leq \gamma_{ij}, \forall (i,j) \in S_m$$

$$\xi_{ij} \geq 0; \gamma_{ij} \geq 0$$

Based on [Joachims 2002]

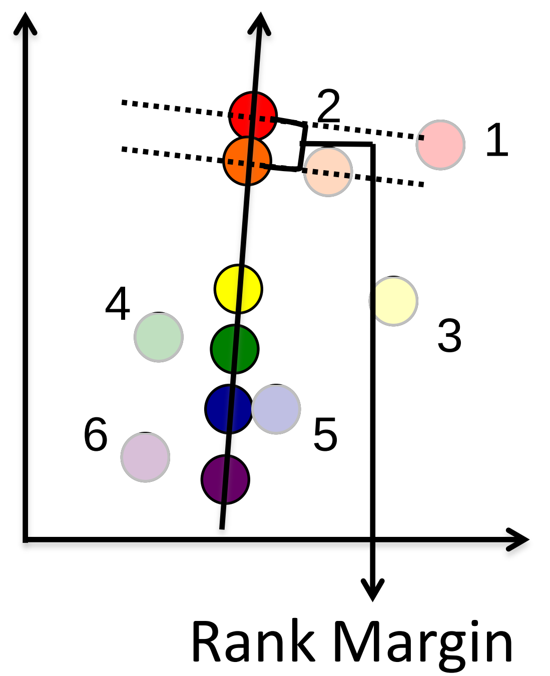
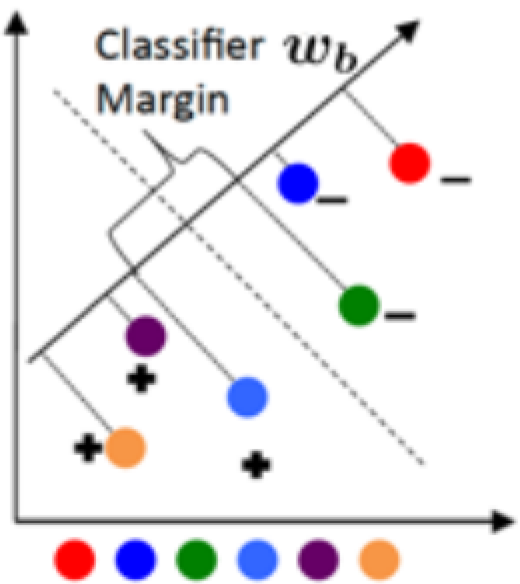


Image $\rightarrow$ Relative Attribute Score

Learning binary attributes v.s. Learning relative attributes

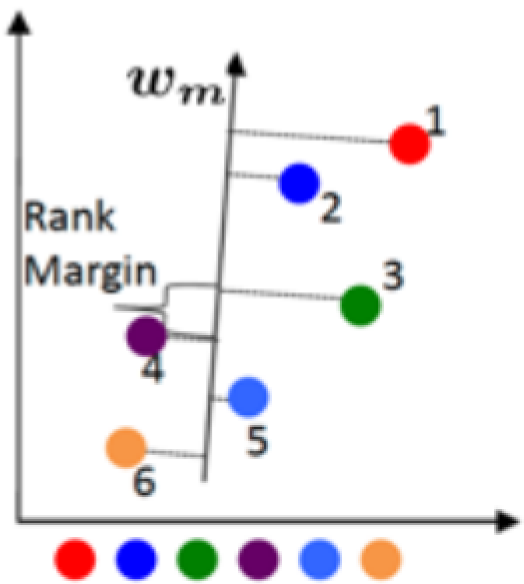
Binary Attributes



Learn decision function

$$d_b(\boldsymbol{x}_i) = \boldsymbol{w}_b^T \boldsymbol{x}_i$$

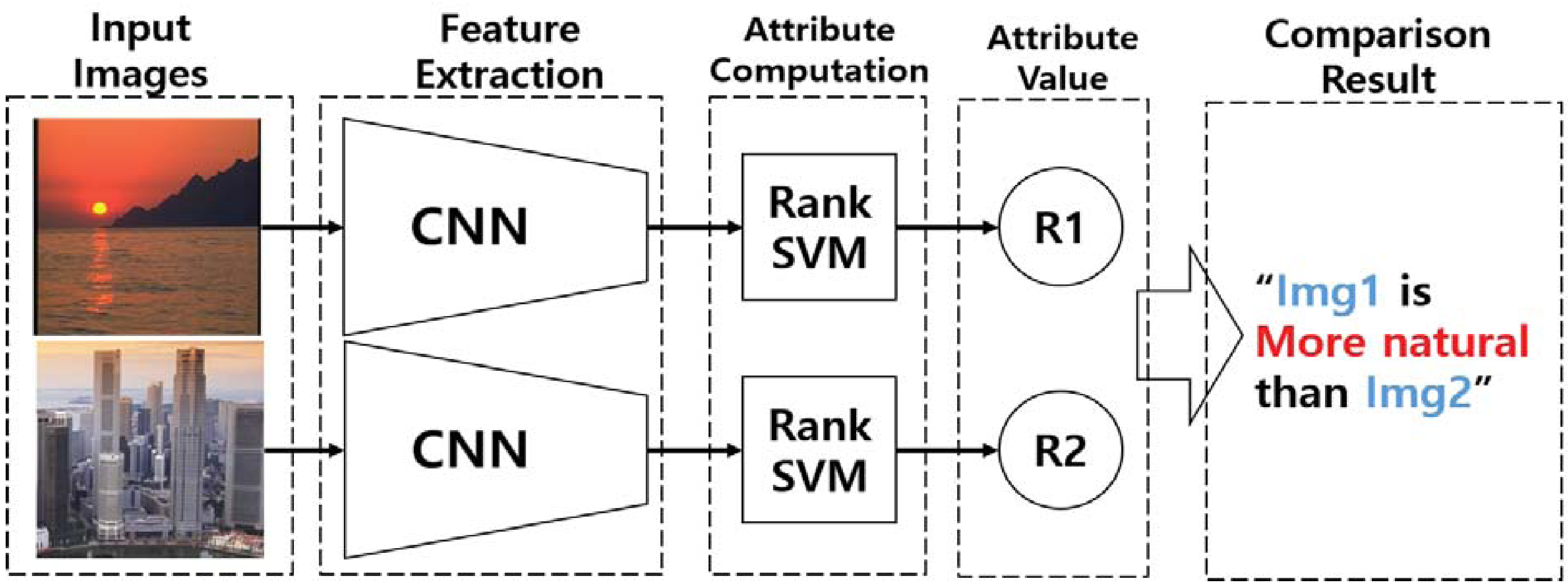
Relative Attributes



Learn ranking function:

$$r_m(\boldsymbol{x}_i) = \boldsymbol{w}_m^T \boldsymbol{x}_i$$

Devi Parikh, Kristen Grauman. Relative Attributes. ICCV 2011.



Dong-Jin Kim, Donggeun Yoo, Sunghoon Im, Namil Kim. Relative Attributes with Deep Convolutional Neural Network. URAI 2015.

RankNet

RankNet

$$\min_{f \in \mathcal{F}_{\text{neural}}} \left[\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \ell_{\text{logistic}}(f, x_i^+, x_j^-) \right]$$

$$\begin{aligned} \ell_{\text{logistic}}(f, x_i^+, x_j^-) &= \log\left(1 + \exp\left(-\left(f(x_i^+) - f(x_j^-)\right)\right)\right) \\ \mathcal{F}_{\text{neural}} &= \text{functions represented by some class of neural networks} \end{aligned}$$

Chris Burges, Tal Shaked, Erin Renshaw. Learning to Rank using Gradient Descent. ICML 2005.

RankNet

➢ Probabilistic Ranking Cost Function

$$o_i \equiv f(\mathbf{x}_i)$$

预测概率： $P_{ij} \equiv \frac{e^{o_{ij}}}{1+e^{o_{ij}}}$ o_{ij}= o(i) - o(j)

真实概率： $\bar{P}_{ij} \equiv \frac{e^{\bar{o}_{ij}}}{1+e^{\bar{o}_{ij}}}$ $\bar{o}_{ij} \equiv \bar{o}_i - \bar{o}_j$

Cross-Entropy Loss：

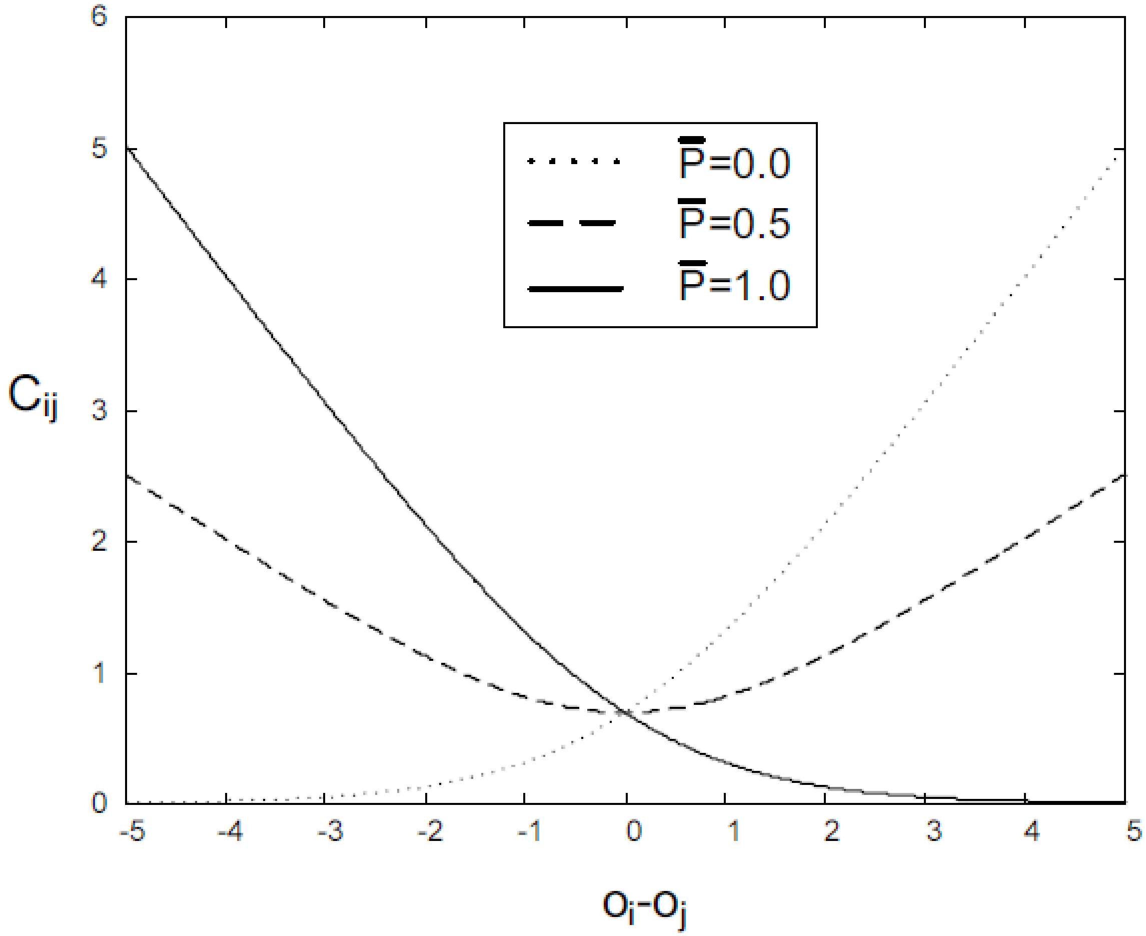
$$C_{ij} \equiv C(o_{ij}) = -\bar{P}_{ij}\log P_{ij} - (1-\bar{P}_{ij})\log(1-P_{ij})$$

$$\Downarrow$$

$$C_{ij} = -\bar{P}_{ij}o_{ij} + \log(1+e^{o_{ij}})$$

RankNet

$$C_{ij} = -\bar{P}_{ij} o_{ij} + \log(1 + e^{o_{ij}})$$

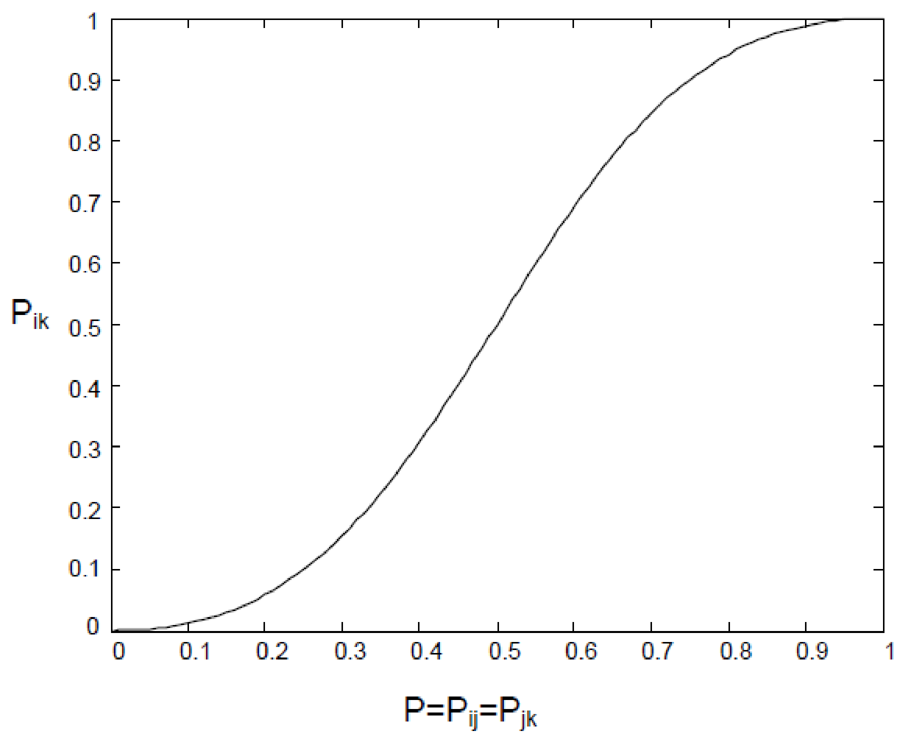


Cost Function for different target probability

RankNet

- Combining Probabilities

$$\bar{P}_{ij} \equiv \frac{e^{\bar{o}_{ij}}}{1+e^{\bar{o}_{ij}}} \qquad \bar{P}_{ik} = \frac{\bar{P}_{ij}\bar{P}_{jk}}{1+2\bar{P}_{ij}\bar{P}_{jk}-\bar{P}_{ij}-\bar{P}_{jk}}$$



$0 < P < 0.5,$ then $\bar{P}_{ik} < P$ $\qquad$ $0.5 < P < 1.0$ then $\bar{P}_{ik} > P$

RankNet

- Back-Propagtion

$$o_i = g^3\left(\sum_j w_{ij}^{32} g^2\left(\sum_k w_{jk}^{21} x_k + b_j^2\right) + b_i^3\right) \equiv g_i^3$$

f(o_i)

$$\frac{\partial f}{\partial b_i^3} = \frac{\partial f}{\partial o_i} g_i'^3 \equiv \Delta_i^3$$

$$\frac{\partial f}{\partial w_{in}^{32}} = \Delta_i^3 g_n^2$$

$$\frac{\partial f}{\partial b_m^2} = g_m'^2\left(\sum_i \Delta_i^3 w_{im}^{32}\right) \equiv \Delta_m^2$$

$$\frac{\partial f}{\partial w_{mn}^{21}} = x_n \Delta_m^2$$

f(o_2-o_1)

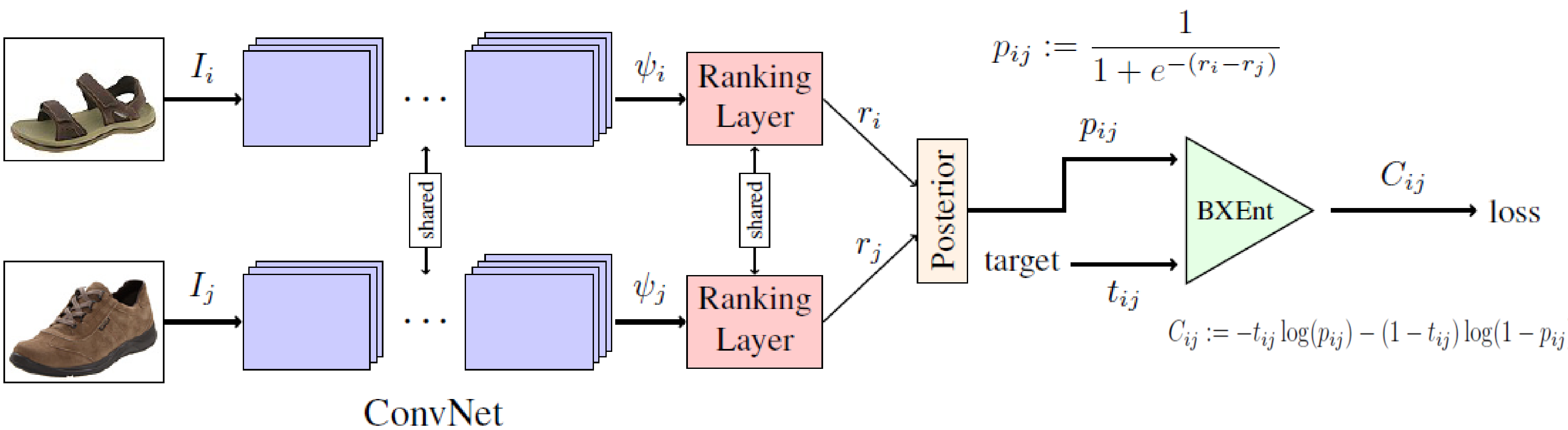
$$\frac{\partial f}{\partial b^3} = f'(g_2'^3 - g_1'^3) \equiv \Delta_2^3 - \Delta_1^3$$

$$\frac{\partial f}{\partial w_m^{32}} = \Delta_2^3 g_{2m}^2 - \Delta_1^3 g_{1m}^2$$

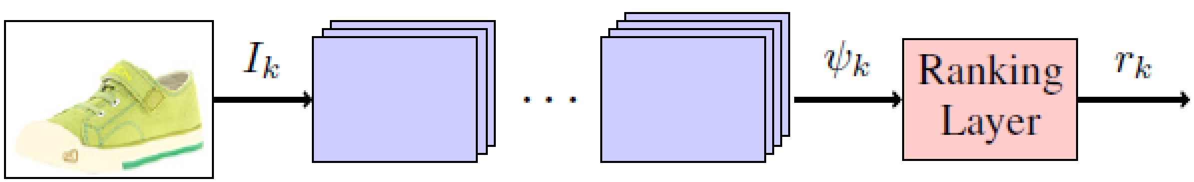
$$\frac{\partial f}{\partial b_m^2} = \Delta_2^3 w_m^{32} g_{2m}'^2 - \Delta_1^3 w_m^{32} g_{1m}'^2$$

$$\frac{\partial f}{\partial w_{mn}^{21}} = \Delta_{2m}^2 g_{2n}^1 - \Delta_{1m}^2 g_{1n}^1$$

Application: Deep Relative Attributes



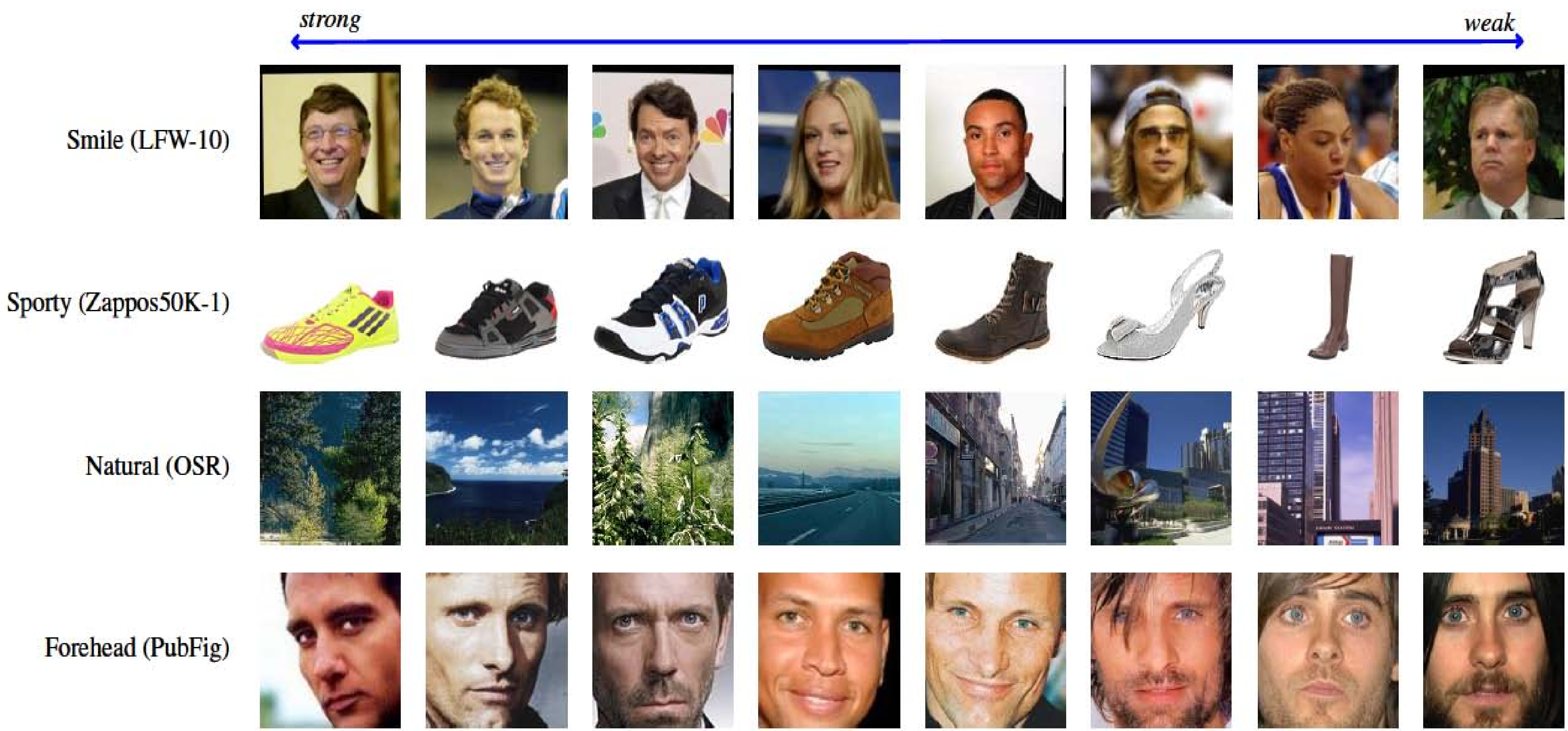
Overview of Training



Test

Yaser Souri, Erfan Noury, Ehsan Adeli-Mosabbeb. Deep Relative Attributes. CVPR 2015.

Deep Relative Attributes



Ranking results of different attributes

Yaser Souri, Erfan Noury, Ehsan Adeli-Mosabbeb. Deep Relative Attributes. CVPR 2015.

Aesthetics Attributes

ID	Attributes	ID	Attributes
1	Post-card like?	10	Memorable vs. Not memorable;
2	Buy this painting?	11	Sky present?
3	Hang-on wall?	12	Clear vs. Cloudy sky;
4	Is aesthetic?	13	Blue vs. Sunset sky;
5	Pleasant vs. Unpleasant;	14	Zoomed in vs. out;
6	Unusual vs. Routine;	15	Top down vs. Side view;
7	Striking vs. Boring colors;	16	Picture of mainly one object vs. Whole scene;
8	High(expert) vs. Poor quality;	17	Single focus vs. Many foci;
9	Attractive vs. Dull photo;		

Is aesthetic:0.80 Memorable:0.28
High quality:0.35 Attractive: 0.23

Is aesthetic:0.00 Memorable:0.76
High quality:0.52 Attractive: 0.50

Phillip Isola, Devi Parikh, Antonio Torralba, and Aude Oliva,“Understanding the intrinsic memorability of images,” in *NIPS,* 2011, pp. 2429–2437.

QA