**MULTI-VIEW DEEP METRIC LEARNING FOR IMAGE CLASSIFICATION**

Dawei Li, Jingjing Tang, Yingjie Tian and Xuchan Ju

{lidewei15, tangjingjing13}@mails.ucas.ac.cn, tyj@ucas.ac.cn

**CONTRIBUTION**
A novel framework that combine the techniques of multi-view learning, multi-view learning and learning are proposed to make image classification. Multiple kinds of features are extracted to obtain information from different sides and deep neural networks make nonlinear transformations on these features to gather similar images and scatter dissimilar images.

**PRELIMINARIES AND NOTATIONS**
Give a multi-view dataset with m training examples from c classes, \( T = \{T_e \in \mathbb{R}^{m \times n} \times Y \}_{e=1}^{V} \), where \( T_e = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \) is the feature set from e-th view and \( y_i \in Y = \{1, 2, \ldots, c\} \) is the label corresponding to each feature.

Metric learning: learning a data-dependent metric to measure similarity more precisely.

Multi-view learning: incorporate the information from different views.

**ACKNOWLEDGEMENTS**
This work has been partially supported by grants from National Natural Science Foundation of China (Nos. 61472390, 11271361, 71331005, and 11226089), Major International (Regional) Joint Research Project (No. 71111017026) and the Beijing Natural Science Foundation (No.1102005).

**REFERENCES**
[2] Xing Eric P and Jordan, Michael I and Russell, Stuart and Ng, Andrew Y. Distance metric learning with application to clustering with side-information In NIPS’02, 2002.

**PROPOSED FRAMEWORK**

\[ V \text{ deep neural networks are constructed, each for a view, to make nonlinear transformation. For each training input } x_{1:v}, \text{ its output of the first layer in the e-th network is } h^L_{1:e}(x_{1:v}) = (W^L_{1:e} x_{1:v}). \]

**OBJECTIVE FUNCTION**

\[
\begin{align*}
\min_{W} J &= \sum_{v=1}^{V} \sum_{i=1}^{m} \alpha_e (d_1(z_{1:v}) + C d_2(z_{2:v})) \\
\end{align*}
\]

**SOLUTION**

Alternative optimization is used to obtain the solution alternately. First, the weight \( \alpha \) is initialized and fixed, then the object function is an unconstrained problem and gradient descent is adopted to solve problem iteratively. The gradient of the objective function with respect to \( W^L_e \) is

\[
\frac{\partial J}{\partial W^L_e} = \alpha_e \sum_{i=1}^{m} \frac{\partial}{\partial W^L_e} (d_1(Cd_2(z_{2:v}))) + \frac{\partial}{\partial W^L_e} (\sum_{i=1}^{m} d_2(z_{1:v})) + \lambda W^L_e
\]

After obtaining the weight matrix \( W \), then \( \alpha \) can be calculated based on the KKT condition,

\[
\alpha = \frac{\mu + \nu^- \kappa - \nu^+ \kappa}{\mu W}
\]

where \( \kappa = (\kappa_1, \ldots, \kappa_V) \in R^V \) and \( \kappa \) is the output of the top hidden layer.

**CLASSIFICATION AND COMPLEXITY**


**PREDICT**

Give a test image with \( V \) views, all of its views will be input to corresponding networks learned and fixed, then the object function is an unconstrained problem and gradient descent is adopted to solve problem iteratively. The gradient of the objective function with respect to \( W^L_e \) is

\[
\frac{\partial J}{\partial W^L_e} = \alpha_e \sum_{i=1}^{m} \frac{\partial}{\partial W^L_e} (d_1(Cd_2(z_{2:v}))) + \frac{\partial}{\partial W^L_e} (\sum_{i=1}^{m} d_2(z_{1:v})) + \lambda W^L_e
\]

After obtaining the weight matrix \( W \), then \( \alpha \) can be calculated based on the KKT condition,

\[
\alpha = \frac{\mu + \nu^- \kappa - \nu^+ \kappa}{\mu W}
\]

where \( \kappa = (\kappa_1, \ldots, \kappa_V) \in R^V \) and \( \kappa \) is the output of the top hidden layer.

**NEAREST NEIGHBORS**


**TABLES**


**FIGURES**


**DIAGRAMS**


**REFERENCES**

[2] Xing Eric P and Jordan, Michael I and Russell, Stuart and Ng, Andrew Y. Distance metric learning with application to clustering with side-information In NIPS’02, 2002.