Efficient Codebook and Factorization for Second-Order Representation Learning

Pierre JACOB¹, David PICARD¹,², Aymeric HISTACE¹, Edouard KLEIN³

1. ETIS, UMR 8051, Université Paris Seine, Université de Cergy-Pontoise, ENSEA, CNRS, F-95000, Cergy, France
2. LIGM, UMR 8049, Ecole des Ponts, UPE, Champs-sur-Marne, France
3. C3N, Pôle Judiciaire de la Gendarmerie Nationale, 5 Boulevard de l’Haut, 95000 Cergy, France

Contact us: {pierre.jacob, aymeric.histace}@ensea.fr, david.picard@enpc.fr

Problems

- How to build rich and compact representations?
- Fine-grained visual tasks ⇒ Second-order pooling
- Unrelated feature aggregation ⇒ Codebook strategies
- Compactness in representation learning ⇒ Deep metric learning (DML)
- What about the drawbacks?
  - Second-order dimensionality is too large for DML
  - Codebook strategies further increase this dimensionality

Joint Codebook-Factorization strategy

- Codebook + second-order pooling increases performances, albeit at a prohibitive cost
- Joint factorization and codebook strategy ⇒ rich and compact representations

Joint Codebook Factorization

Raw representation

- We duplicate the codebook assignment for symmetry purpose:
  \[ y = h(x) \odot x \odot h(x) \odot x \]

Raw projection

- The i-th dimension is computed as follows:
  \[ z_i = \langle w_i ; h(x) \odot x \odot h(x) \odot x \rangle \]

First factorization

- Rank-one factorization to split each pair of codebook assignment and feature:
  \[ z_i = \langle p_i ; h(x) \odot x \rangle \langle q_i ; h(x) \odot x \rangle \]

Second factorization

- Multi-rank factorization which generalize intra-projection in codebook strategies:
  \[ p_i = \sum_j e^{(j)} \odot u_{i,j} \quad \text{and} \quad q_i = \sum_j e^{(j)} \odot v_{i,j} \]
  \[ z_i = \left(h(x)^\top U_i^\top x\right) \left(h(x)^\top V_i^\top x\right) \]

Sharing projectors

- Entries in projection matrices can be shared between codebook entries:
  \[ z_i = \left(h(x)^\top A U_i^\top x\right) \left(h(x)^\top B V_i^\top x\right) \]

Results

Comparison to the state-of-the-art on three image retrieval datasets

- Decrease in parameters can be handled by sharing projectors at the cost of a drop in performances

Conclusion

- Codebook strategies further improve second-order pooling
- Joint codebook-factorization greatly reduces the representation size with fewer loss in performances

Acknowledgments

We would like to acknowledge COMUE Paris Seine University, University of Cergy-Pontoise and M2M Factory for their financial and technical support.