SVM Speedup Project

Problem Statement: The aim is to efficiently evaluate linear SVM on large image database. Given a query hyperplane (SVM weight vector) $\mathbf{w}$, the goal is to retrieve top-k highest scored images with respect to $\mathbf{w}$. The SVM scores are computed as $\mathbf{w}^T\mathbf{x_i}, i = 1 \rightarrow n$, where $\mathbf{x_i} \in \Re^d$ is the feature vector corresponding to i^{th} database image and n is the database size.

Dimensionality Reduction Approach: As described in [1], we consider models that learn the mapping into a low dimensional feature space where both hyperplanes and images are represented. The low dimensional embeddings allow fast computation of the scores with small memory usage. We jointly learn mappings for both hyperplanes $\phi_h(\mathbf{w})$ and images $\phi_x(i)$ that map into a low dimensional joint space $\Re^p$.

$$\phi_h(\mathbf{w}) : \Re^d \rightarrow \Re^p$$

$$\phi_x(i) : \Re^d \rightarrow \Re^p$$

The mapping for hyperplanes is defined by $\phi_h(\mathbf{w}) = \mathbf{w}^T\mathbf{U}$ and for images it is defined by $\phi_x(i) = \mathbf{V}^T\mathbf{x_i}$. All the database images $\mathbf{x_i}$ are ranked according to the following scoring function:

$$f_i(\mathbf{w}) = \phi_h(\mathbf{w})^T\phi_x(i) = \mathbf{w}^T\mathbf{U}\mathbf{V}^T\mathbf{x_i} \tag{1}$$

Let $f(\mathbf{w}) \in \Re^n$ be a vector function providing a score for each of the images, where $f_i(\mathbf{w})$ is the score of the i^{th} database image. We use sampling method with stochastic gradient descent (SGD) for optimizing precision at k for our model.

1. Pick a random labeled example $(\mathbf{w}, \mathbf{x})$, where $\mathbf{x}$ is highest scored image with respect to $\mathbf{w}$.
2. For the chosen pair $(\mathbf{w}, \mathbf{x})$, select a violating image $\mathbf{x}'$ such that $1 + f_{\mathbf{x}'}(\mathbf{w}) > f_{\mathbf{x}}(\mathbf{w})$

The ranking error function based on the triplet $(\mathbf{w}, \mathbf{x}, \mathbf{x}')$ is defined as:

$$err(f(\mathbf{w}), \mathbf{x}, \mathbf{x}') = L(rank_{\mathbf{x}}(f(\mathbf{w})))|1 - f_{\mathbf{x}}(\mathbf{w}) + f_{\mathbf{x}'}(\mathbf{w})|_+ \tag{2}$$

where $|t|_+$ is the positive part of $rank_{\mathbf{x}}(f(\mathbf{w}))$ is the margin-penalized rank of $\mathbf{x}$:

$$rank_{\mathbf{x}}(f(\mathbf{w})) = \sum_{i \neq \mathbf{x}} I(1 + f_i(\mathbf{w}) > f_{\mathbf{x}}(\mathbf{w})) \tag{3}$$

where I is the indicator function and $L(\cdot)$ transforms this rank into loss:

$$L(t) = \sum_{j=1}^{t} 1/j \tag{4}$$

- *Speedup:* The computational cost of evaluating SVM ($\mathbf{w}^T\mathbf{x_i}$) over image database of size n with d dimensional feature representation is $\mathcal{O}(nd)$. Instead if we use $f_i(\mathbf{w})$ (Equation 1) for the same, then the computational cost will be $\mathcal{O}(d+n)p$, where $p << d$, is the dimensionality of the joint space to which both hyperplanes and images are mapped to. Thus for large n, we get approximate speedup of d/p.

- *Comparison:* Since, the approach is based on dimensionality reduction we should compare the proposed method with other dimensionality reduction techniques like PCA.

Scoring Independent of n: The speedup using only dimensionality reduction techniques will be very low. For example, for large dimensional feature representation like histogram of bag of words, the feature vectors are usually sparse. Hence the actual speedup is d'/p, where $d' << d$, is the average number of non-zero features. Thus, to get a significant speedup we need to index images such that the SVM evaluation becomes independent of n. One approach is to come up with the learning model which map both the hyperplanes and images onto binary feature space [2, 3]. We can use the topk-pruning technique [4] for fast svm evaluation on the binary features.

References

[1] Weston, J., Bengio, S., Usunier, N.: Wsabie: scaling up to large vocabulary image annotation. IJCAI (2011)

[2] Norouzi, M., Fleet, D.: Minimal loss hashing for compact binary codes. ICML (2011)

[3] Norouzi, M., Fleet, D., Salakhutdinov, R.: Hamming distance metric learning. In: NIPS. (2012)

[4] Rastegari, M., Fang, C., Torresani, L.: Scalable object-class retrieval with approximate and top-k ranking. In: ICCV. (2011)