Scalable Multitask Representation Learning for Scene Classification

Maksim Lapin[1,2], Bernt Schiele[1], Matthias Hein[2]

1: Max Planck Institute for Informatics, Saarbrücken, Germany 2: Department of Mathematics and Computer Science, Saarland University

GitHub / mlapin

Contributions

- state of the art on SUN397 benchmark[1]
  - **49.5%** (SIFT & LCS ⇨ Fisher Vector ⇨ MTL)
- consistent improvement over standard one-vs-all single task learning (STL)
  - w/ and w/o color cues
  - **5..50** training examples per class
  - top-K accuracy for **all** K
- scalability to Fisher Vector features
  - **260 000** dimensions, dense

Scene Classification

- previous state of the art[2] (**47.2%**)
  - Fisher Vector on SIFT & LCS (color feature)
  - independent one-vs-all SVMs
- SUN397 challenges
  - groups of related (ambiguous) classes
  - ≤50 training examples per class

combined with high dimensional FV features ⇒ overfitting

- existing relations between classes could be exploited

industrial area

- discrimination between visually similar classes is hard (also for humans)
- forcing a one-vs-all classifier to separate 'nuclear power plants' from 'power plants' may lead to increased overfitting
- instead, our method separates classes in a lower dimensional subspace, which is learned jointly for all scene categories

power plant

specialization

nuclear power plant

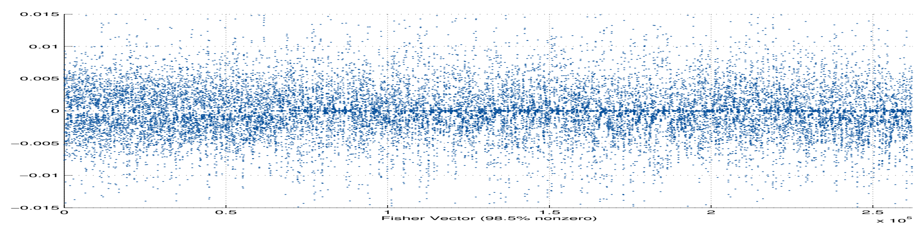
Multitask Representation Learning — MTL-SDCA

- effective regularization (lower dimensional subspace)
- joint learning of mapping U (multitask learning - MTL)

$$\underbrace{\min_U \frac{1}{T}\sum_{t=1}^{T}}_{\text{multitask learning}} \underbrace{\left[\min_{w_t} \frac{1}{n}\sum_{i=1}^{n} \max\{0, 1 - y_{ti}\langle w_t, \underbrace{Ux_i}_{\text{subspace}}\rangle\} + \frac{\lambda}{2}\|w_t\|^2\right]}_{\text{one-vs-all SVM}} + \frac{\mu}{2}\|U\|_F^2$$

Algorithm

1. start with an image representation x_i (e.g., Fisher Vector, but could be any)



2. train one-vs-all SVMs on x_i (first layer, initialization for MTL)

standard approach up to this point

3. stack learned predictors into U_0
4. iterate (multitask learning)
   - train SVMs on Ux_i
   - update U ⇐ crucial step

prediction cost is effectively the same as STL since additional product is low dim.

5. final prediction:
   - $\arg\max_{t=1,\ldots,T} \langle w_t^\star, U^\star x\rangle$

Implementation Details (code on GitHub!)

- adapt SDCA solver[3] (Stochastic Dual Coordinate Ascent)
  - no primal variables, all in dual
  - learning U via SDCA-variant
- both subproblems via SDCA (hence MTL-SDCA)
- use precomputed kernels (dual optimization: n=20K ≪ d=260K)
- closed-form updates, also for U

$$\alpha_{ti}^{(s)} = \begin{cases} \max\left(0, \min\left(C, \alpha_{ti}^{(s-1)} + h\right)\right) & \text{if } y_{ti} = +1, \\ \max\left(-C, \min\left(0, \alpha_{ti}^{(s-1)} - h\right)\right) & \text{if } y_{ti} = -1, \end{cases}$$

$$h = \frac{1 - y_{ti} K_i^\top A M_t}{K_{ii} M_{tt}}, \; C = \frac{1}{\mu n T}, \; K = X^\top X, \; M = U_0^\top U_0$$

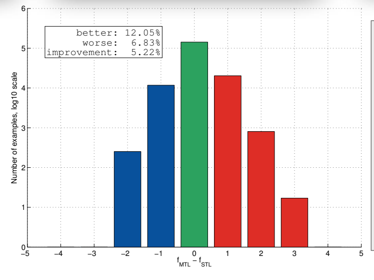
Runtime Comparison*

	MTL	STL	Overhead
SDCA training	25min	2min	x11
+kernels $+U_0$	33min	8min	x4
+image representation	6.7h	6.2h	x1.07

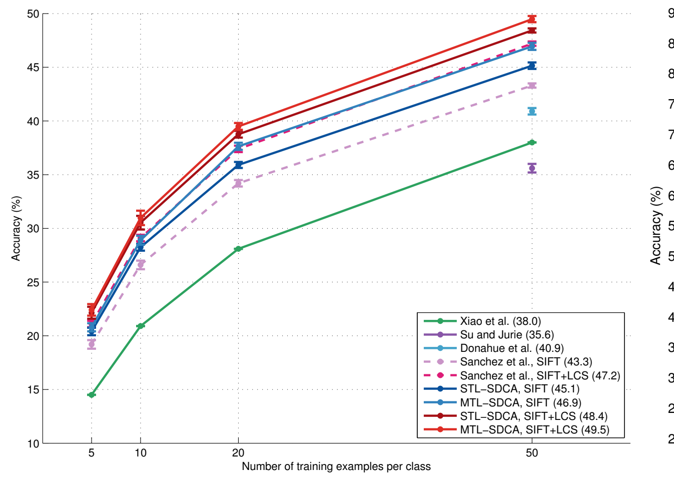
*further details can be found in the supplementary material

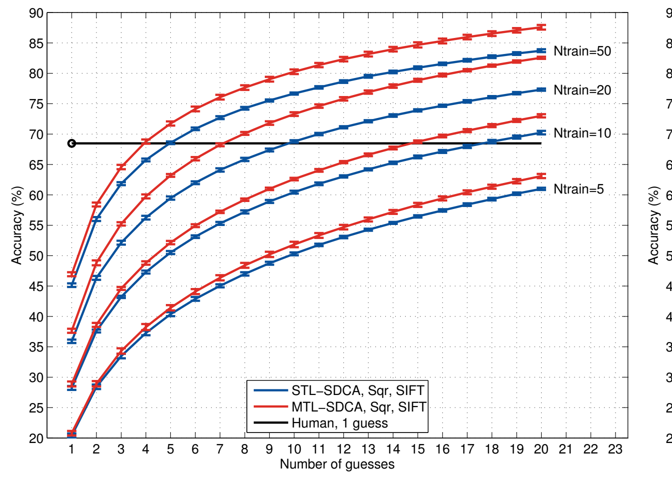
Experiments

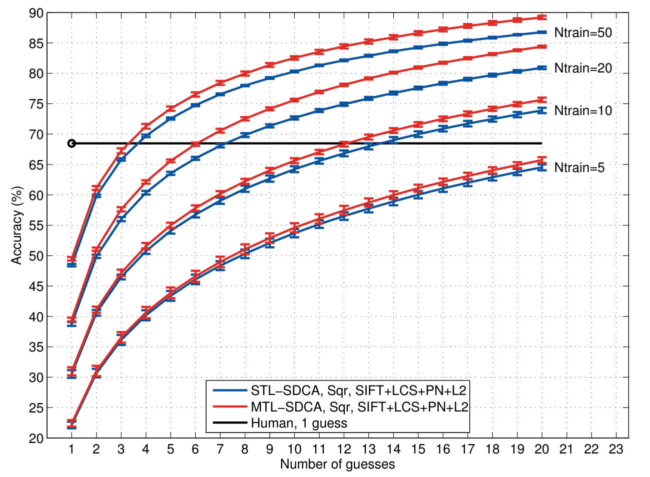
- evaluation on SUN397
  - FV fine-tuning +1.2%
  - top-K accuracy (top-5/15: +3.7%/+5%)
- sanity check on MNIST/USPS (improvement over STL, on par w/ another MTL)



MTL-SDCA exhibits a small tendency to produce more human-like predictions







Left: SUN397 state of the art. Middle: STL vs MTL, SIFT only (top-K accuracy). Right: STL vs MTL, SIFT and color.

References

[1] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from Abbey to Zoo. CVPR'10.

[2] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the Fisher vector: theory and practice. IJCV'13.

[3] S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. JMLR'13.

https://github.com/mlapin/cvpr14mtl

Conclusion

- effective MTL regularization consistently improves over STL
- achieves state of the art results
- scales to dense high dimensional image representation (Fisher Vector)