Table 1: **Summary Statistics of the Three Datasets Used in the Experiments.**

Statistics	ImageNet	Product Descriptions	Product Images
Task	image annotation	product categorization	image annotation
Number of Training Documents	2518604	417484	417484
Number of Test Documents	839310	60278	60278
Validation Documents	837612	105572	105572
Number of Labels	15952	18489	18489
Type of Documents	images	texts	images
Type of Features	visual terms	words	dense image features
Number of Features	10000	10000	1024
Average Feature Sparsity	97.5%	99.6%	0.0%

Table 2: **Flat versus Tree Learning Results** Test set accuracies for various tree and non-tree methods on three datasets. Speed-ups compared to One-vs-Rest are given in brackets.

Classifier	Tree Type	ImageNet	Product Desc	Product Images
One-vs-Rest	None (flat)	2.27% [1×]	37.0% [1×]	12.6% [1×]
Filter Tree	Filter Tree	0.59% [140×]	14.4% [125×]	0.73% [120×]
Conditional Prob. Tree (CPT)	CPT	0.74% [41×]	26.3% [45×]	2.20% [115×]
Independent Optimization	Random Tree	0.72% [60×]	21.3% [59×]	1.35% [61×]
Independent Optimization	Learnt Label Tree	1.25% [60×]	27.1% [59×]	5.95% [61×]
Tree Loss Optimization	Learnt Label Tree	2.37% [60×]	39.6% [59×]	10.6% [61×]

Table 3: **Label Embeddings and Label Embedding Tree Results**

		ImageNet			Product Images		
Classifier	Tree Type	Accuracy	Speed	Memory	Accuracy	Speed	Memory
One-vs-Rest	None (flat)	2.27%	1×	1.2 GB	12.6%	1×	170 MB
Compressed Sensing	None (flat)	0.6%	3×	18 MB	2.27%	10×	20 MB
Seq. Convex Embedding	None (flat)	2.23%	3×	18 MB	3.9%	10×	20 MB
Non-Convex Embedding	None (flat)	2.40%	3×	18 MB	14.1%	10×	20 MB
Label Embedding Tree	Label Tree	2.54%	85×	18 MB	13.3%	142×	20 MB

Embedding and Embedding Tree Approaches In Table 3 we compare several label embedding methods: (i) the convex and non-convex methods from Section 5; (ii) compressed sensing; and (iii) the label embedding tree from Section 3.2. In all cases we fixed the embedding dimension $d_e = 100$. The results show that the random embeddings given by compressed sensing are inferior to learnt embeddings and Non-Convex Embedding is superior to Sequential Convex Embedding, presumably as the overall loss which is dependent on both W and V is jointly optimized. The latter gives results as good or superior to One-vs-Rest with modest computational gain (3× or 10× speed-up). Note, we do not detail results on the product descriptions task because no speed-up is gained there from embedding as the sparsity is already so high, however the methods still gave good test accuracy (e.g. Non-Convex Embedding yields 38.2%, which should be compared to the methods in Table 2). Finally, combining embedding and label tree learning using the "Label Embedding Tree" of Section 3.2 yields our best method on ImageNet and Product Images with a speed-up of 85× or 142× respectively with accuracy as good or better than any other method tested. Moreover, memory usage of this method (and other embedding methods) is significantly less than One-vs-Rest.

6 Conclusion

We have introduced an approach for fast multi-class classification by learning label embedding trees by (approximately) optimizing the overall tree loss. Our approach obtained orders of magnitude speedup compared to One-vs-Rest while yielding as good or better accuracy, and outperformed other tree-based or embedding approaches. Our method makes real-time inference feasible for very large multi-class tasks such as web advertising, document categorization and image annotation.

Acknowledgements

We thank Ameesh Makadia for very useful discussions.