Scalable Training of Interpretable Spatial Latent Factor Models

Stephan Zheng
Caltech
stzheng@caltech.edu

Yisong Yue
Caltech
yyue@caltech.edu

Abstract

We study the problem of learning interpretable spatial latent factor models. In contrast to other types of latent factor models, the goal of learning interpretable spatial models is to learn a set of latent factors that is not only predictive but also captures coherent spatial semantics. We present a multi-stage meta-algorithm for reliably training such models that scales to very fine-grained spatial models with higher-order interactions (i.e., tensor latent factor models). We further propose a family of instantiations of this meta-algorithm for fast and efficient training.

1 Introduction

In this paper, we study the problem of learning spatial latent factor models that can capture semantically cohesive spatial patterns. One convenient property of spatial models is that they are straightforward to visualize and interpret. For instance, Figure 1 shows two spatial models describing shooting patterns in basketball, one of which is interpretable while the other is not. We utilize a simple yet general class of spatial models defined over fine-grained discretizations of the raw data, which requires making minimal assumptions regarding the functional form of the spatial patterns.

Due to the non-convex nature of the training objective, it can be challenging to train an accurate yet interpretable model. Figure 1 shows two example latent factors learned via gradient descent: the left using a good initialization and the right using random. As such, selecting a proper initialization is key.

To address this issue, we initialize using the factorization of a trained "full-rank" or un-factorized model, which has been shown to be effective for capturing cohesive behavioral spatial semantics such as for basketball game play [1, 2], and was the initialization method used in Figure 1 (left). Of course, training a full-rank model can be extremely expensive, despite the learning objective being convex. For instance, an un-factorized 3-tensor model has complexity that scales cubically with the spatial granularity.

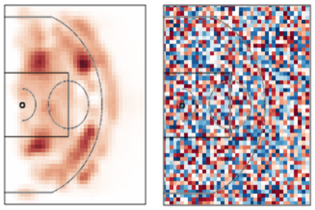

Figure 1: Showing interpretable (left) and uninterpretable (right) latent factors. The left was trained using a good initialization, whereas the right was trained using random initialization.

Our approach is motivated by the insight that a coarse-grained spatial model has relatively low accuracy, but trains fast and provides a good initialization for a more fine-grained model. We formalize this intuition in a multi-stage training procedure that uses iteratively more fine-grained models, and thus allows for training a relatively coarse un-factorized model to initialize the latent factor model.

Our approach bears affinity to other multi-stage meta-algorithms (e.g., [3]). Key design decisions include: (a) deciding when to terminate the current stage; and (b) choosing the parameterization of the model in the next stage. For the latter, we choose a straightforward coarse-to-fine parameterization. For the former, we set the termination condition via a notion of *spatial entropy*. We demonstrate that our method trains faster than fixed-resolution approaches and reliably finds interpretable solutions for a practically relevant problem. In summary, our main preliminary results are:

- Our method converges faster than training at a fixed spatial resolution from the beginning.
- Our method reliably and efficiently yields an interpretable spatial latent factor model. Our method easily scales to higher-order tensor latent factor models.

2 The Learning Problem

We consider a multi-task binary classifier that predicts a label per task $a \in \mathcal{A}$ given an input instance x. For example, the model can predict whether basketball player a in spatial position x would shoot the ball. More generally, our approach can be applied to any model with at least one spatial component. For each task a and spatial input x, we aim to learn an expressive yet compact scoring function $f_a(x)$, after which classification is simply $h_a(x) = \texttt{sign}(f_a(x))$. One common approach is to use tensor models that can capture higher-order interactions:

$$f_a(x) = \sum_{b,c} \psi_b(x)\phi_c(x)F_{abc} + b_a = \sum_{b,c} \psi_b(x)\phi_c(x)\left(A \otimes B \otimes C\right)_{abc} + b_a, \tag{1}$$

where $\psi(x), \phi(x)$ are discretization feature vectors over (dimensions of) x, F is a 3-tensor of parameters, and b_a is a bias unit for task a. We further assume that F is low-rank and decomposes into a PARAFAC product [4], where A, B, C are matrix latent factors and k indexes the latent dimension. Since ψ and ϕ define discretizations over x, the granularity of ψ and ϕ define the granularity of F.

We train our spatial tensor model (1) on a multitask dataset $D = \{x_i, y_{i,a} : j \in \mathcal{I}, a \in \mathcal{A}(i)\}$, where every x_i is associated with a binary task label $y_{i,a} \in \{0, 1\}$. We minimize the standard trade-off between prediction loss and regularization:

$$\min_{\Omega} \mathcal{L}(\Omega) + \mathcal{R}(\Omega), \quad \Omega = \{A, B, C, b\}, \tag{2}$$

where Ω denotes all the parameters, $\mathcal{R}(\Omega)$ is a regularization and $\mathcal{L}(\Omega)$ is the standard log loss:

$$\mathcal{L} = \sum_{i \in I} \sum_{a \in \mathcal{A}(i)} \left(-y_{i,a} f_a(x_i) + \log\left(1 + e^{y_{i,a} f_a(x_i)}\right)\right), \tag{3}$$

where $\mathcal{A}(i)$ denotes the set of tasks active in example i (e.g., only certain basketball players appear in a given frame). In order for A, B, C to be smooth, we use L_2 regularization on all factors.

3 Multi-Stage Optimization Approach

We now describe our meta-algorithm for training interpretable spatial latent factor models. We will assume as axiomatic that local optimization from a "good" initialization of the latent factor model will lead to an interpretable local optimum, and that approximately training the full-rank version of F and then factorizing will provide such a good initialization.[1]

The principal issue that we address is the fact that training spatial models can be very expensive, especially the un-factorized F used for initialization. Our multi-stage approach builds upon the insight that one can very quickly train a coarse-grained spatial model, which can be used to initialize a more fine-grained model. A high-level intuition of our approach is depicted in Figure 2.

We choose n levels of strictly increasing spatial resolutions, $R_1 < \ldots < R_n$, and denote the multi-resolution features by $\Psi = \left\{\psi^i\right\}_{1 \ldots n}$, e.g., histogram features ψ^i using a spatial grid with R_i cells. The MMT-SE procedure (Algorithm 1) starts with a full-rank model at spatial resolution R_1 and trains using SGD-SE (Algorithm 2) until some termination condition is met.[2] The resolution is then increased $R_1 \rightarrow R_2$, typically by a multiplicative scale factor. We iteratively repeat this process in the MMT-SE procedure until some resolution R_p, after which we factorize the full-rank F as initialization for the latent factor model (denoted TensorFactors in Algorithm 1).[3] We can use any tensor factorization approach to compute the factorization (e.g., [5, 6]). Afterwards, we continue training the factorized low-rank model up to the final resolution R_n.

[1] This assumption has been empirically validated (cf. [1, 2]), and one interesting direction for future work would be to formally characterize this property.

[2] In principle, one could use any other optimization subroutine.

[3] One natural choice of R_p is when the memory requirement of R_{p+1} exceeds the system's capacity.

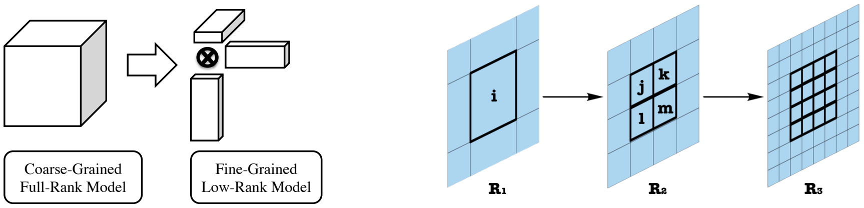

Figure 2: Left: depicting the start and end stages, where training initially focuses on a coarse-grained full-rank model, and concludes on a fine-grained latent factor model. Right: depicting the intermediate stages whereby the model is made iteratively more fine-grained. See Algorithm 1 and Algorithm 2 for more details.

Algorithm 1 MMT-SE : Memory-efficient multi-resolution training with spatial entropy control

```
Input: Tensor$^0$, fixed weights $\Omega$, data-set $D$, features $\Psi$.
for each resolution $R_i \in \{R_1, \ldots, R_p\}$ do
    SGD-SE $\left(\textbf{Tensor}^i; \Omega\right)$
end for
TensorFactors$^p$ = TENSOR-FACTORIZE($F^p$)
for each resolution $R_i \in \{R_{p+1}, \ldots, R_n\}$ do
    SGD-SE $\left(\textbf{TensorFactors}^i; \Omega\right)$
end for
return TensorFactors$^n$
```

Algorithm 2 SGD-SE : Stochastic gradient descent with spatial entropy control

```
Input: Weights$^i$, fixed weights $\Omega$, data-set $D$, features $\Psi$.
while condition (5) not true do
    Mini-batch gradient descent step on Weights
end while
FINEGRAIN($\alpha^i$)
return Weights$^{i+1}$
```

3.1 Spatial Entropy Termination Criterion

Given the structure of Algorithm 1, the key technical question is how to set the termination condition for each stage. Intuitively, one simple termination condition is when the optimization problem at the present resolution converges (which mimics the termination condition for iterative training of sparse models [3]). However, since the model at each resolution is used to initialize the training for the subsequent resolution, it need not be that the converged model at the current resolution will be the best initialization for the next resolution (i.e., training might be overfitting to the coarser resolution).

We therefore use a notion of *spatial entropy* of the gradient distributions. Intuitively, spatial resolution R is too coarse when the training data prefers much more fine-grained curvature, as evidenced by substantial disagreement in the gradients for resolution R. More formally, for a function f defined using resolution R and trained on $X = \{x_i\}$, we define its spatial entropy $S^R\left(f, \mathcal{I}(X)\right)$ as the entropy of the empirical distribution $P^R\left(f, \mathcal{I}(X)\right)$ of gradients $\{\nabla f(x_i) : i \in \mathcal{I}(X)\}$:

$$S^R\left(f, \mathcal{I}(X)\right) = -\sum_{i \in \mathcal{I}(X)} P^R\left(f, \mathcal{I}(X)\right) \log P^R\left(f, \mathcal{I}(X)\right). \tag{4}$$

Here $\mathcal{I}(X)$ means we consider the gradients from a set of gradient evaluations, e.g. a rolling window of gradients during mini-batch training. A straightforward termination condition is then when the spatial entropy of the model starts to increase more than a margin Δ, i.e. at iteration t

$$S_t^R > S_{t-1}^R + \Delta, \tag{5}$$

indicating that the gradients of the current discretization are increasingly disagreeing with each other.

4 Benchmark Experiments: Shot prediction on basketball tracking data

We validate our approach with basketball shot prediction, where the tasks $a \in \mathcal{A}$ correspond to different basketball players and our multi-task model predicts whether player a will shoot under spatial conditions x. For this prediction problem, we used a large player tracking dataset that includes hundreds of players and covers millions of game frames captured during competitive basketball game play. For every frame i, the binary labels $y_{i,a} \in \{0, 1\}$ indicate whether the player will shoot.

In this problem, we use occupation features $\psi(x_i), \phi(x_i)$ for the ball handler and defenders respectively. At full resolution, the left half court is discretized using a 50×40 grid of $R = 2000$ cells of size 1×1, while lower resolutions use $2 \times 2, 5 \times 5$ and 10×10 cells. Furthermore, we only consider defenders in a 12×12 grid ($R_d = 144$) around the ball handler, with 1×1 and 2×2 cells. A similar data representation was used in [2].

4.1 Results

Recall that our approach is based on first training a full-rank model to initialize the factorized model. We compare our multi-stage approach with learning at a fixed resolution. Moreover, we compare two termination criteria: fine-graining when the loss has converged versus spatial entropy control. For the latter, we used condition (5) with $\Delta = 0.005$ to determine when to fine-grain. Figure 3 shows our results. The left plot shows the results for the full-rank model, which is usually the more computationally intensive part. We see that our multi-stage approach dramatically outperforms (by multiple orders-of-magnitude) a naive approach which only uses the finest resolution.[4] Moreover, fine-graining controlled by spatial entropy outperforms, by an order-of-magnitude, using the loss as a termination criterion. The right plot shows the performance of our method on the latent factor model after initializing using a factorized version of the previous stage. We see that the learning objective continues to decrease as the learning problem enters what is essentially a fine-tuning phase.[5]

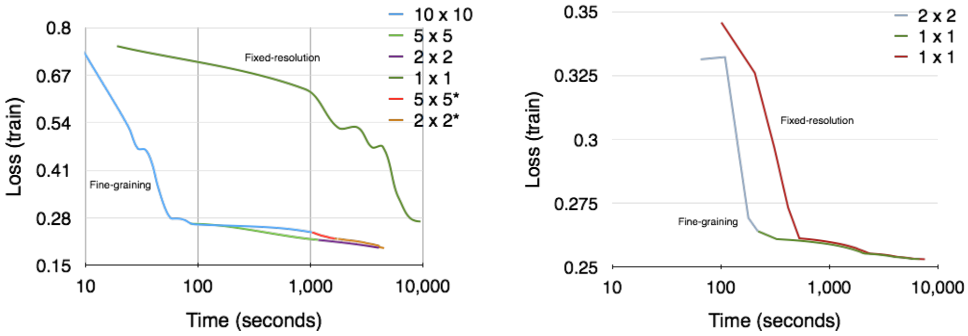

*Figure 3: Left: training a full-rank tensor model using iterative fine-graining. Fine-graining is controlled by checking (5) per minibatch. Baseline 1: training a fixed-resolution fine-grained model is orders of magnitude slower. Baseline 2 (indicated by *): fine-graining when the loss has converged, gives slower convergence. Right: training a factored tensor model using iteratively fine-graining outperforms a fixed-resolution approach.*

5 Discussion

We have presented a fast and memory-efficient multi-resolution approach to train interpretable spatial latent factor models defined over fine-grained discretizations of the raw spatial data. Our method easily accommodates higher-order interactions such as tensor latent factor models. Our preliminary results demonstrate substantial speed-up in training over conventional gradient descent approaches. Our results also suggest several interesting directions for further study:

- Can we formally characterize the type of interpretable models that are obtainable through this type of initialization scheme? For instance, many types of non-convex learning problems can be well-solved by first obtaining a good initialization (cf. [7]).
- Can we formally characterize how spatial entropy captures the correct termination condition? That is, can we prove that as spatial entropy increases, training in the coarser resolution no longer guarantees progress in the finer resolution? Is there a more refined version of spatial entropy that can lead to even more speedup?
- Can we formally characterize how much the multi-stage approach can speed up training relative to baseline gradient descent approaches?
- Can we extend this approach to deal with adaptive or non-uniform multi-resolution settings and enable even further speedup?

[4] Note also that for more complex models, the naive approach would not even fit in main memory.

[5] Note the absolute objective of the un-factorized model is lower than the latent-factor model because the un-factorized model has more degrees of freedom and is overfitting to the training data.

References

[1] Andrew Miller, Luke Bornn, Ryan Adams, and Kirk Goldsberry. Factorized point process intensities: A spatial analysis of professional basketball. In *International Conference on Machine Learning (ICML)*, 2014.

[2] Yisong Yue, Patrick Lucey, Peter Carr, Alina Bialkowski, and Iain Matthews. Learning Fine-Grained Spatial Models for Dynamic Sports Play Prediction. In *IEEE International Conference on Data Mining (ICDM)*, 2014.

[3] Tyler B Johnson and Carlos Guestrin. Blitz: A principled meta-algorithm for scaling sparse optimization. In *International Conference on Machine Learning (ICML)*, 2015.

[4] Evrim Acar, Daniel Dunlavy, Tamara Kolda, and Morten Mørup. Scalable tensor factorizations with missing data. In *SIAM Conference on Data Mining (SDM)*, 2010.

[5] Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, 2009.

[6] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Neural Information Processing Systems (NIPS)*, 2001.

[7] Arun Tejasvi Chaganty and Percy Liang. Spectral experts for estimating mixtures of linear regressions. In *International Conference on Machine Learning (ICML)*, 2013.