Supplementary Material: Gaussian Mixture Latent Vector Grammars

Yanpeng Zhao, Liwen Zhang, Kewei Tu
School of Information Science and Technology,
ShanghaiTech University, Shanghai, China
{zhaoyp1,zhanglw1,tukw}@shanghaitech.edu.cn

Abstract

This supplementary material contains the following contents. (1) The derivation of the gradient formulation (Equation 12 in the paper). (2) The general idea of calculating analytic gradients for all the parameters in Gaussian Mixture Latent Vector Grammars (GM-LVeGs). (3) Algorithmic complexity and running time. (4) Statistics of the data used for part-of-speech (POS) tagging and constituency parsing. (5) Additional experimental results and analysis of GM-LVeGs.

1 Derivation of Gradient Formulations

Given a training dataset $D = \{(T_i, \mathbf{w}_i) \,|\, i = 1, \ldots, m\}$ containing m samples, we minimize the negative log conditional likelihood during learning of GM-LVeGs:

$$\mathcal{L}(\Theta) = -\log \prod_{i=1}^{m} P(T_i|\mathbf{w}_i; \Theta)\,, \tag{1}$$

where T_i is the gold parse tree with unrefined nonterminals for the sentence $\mathbf{w}_i$, and Θ is the set of parameters in GM-LVeGs.

We define t as a parse tree with nonterminal subtypes, denote by $f_r(t)$ the number of occurrences of the unrefined rule r in the unrefined parse tree that is obtained by replacing all the subtypes in t with the corresponding nonterminals, and use $\mathbf{r}$ to represent a fine-grained production rule of r, which is represented by the concatenation of the latent vectors of the nonterminals in r.

The weight of t is defined as:

$$s_t = \prod_{\mathbf{r} \in t} W_r(\mathbf{r})\,. \tag{2}$$

The weight of T, a parse tree with unrefined nonterminals, is defined as:

$$s_T = \int_{t \sim T} s_t \, dt\,, \tag{3}$$

where $t \sim T$ indicates that t is a parse tree with nonterminal subtypes that can be converted into a parse tree T by replacing its nonterminal subtypes with the corresponding nonterminals. The weight of a sentence $\mathbf{w}$ is defined as:

$$s_{\mathbf{w}} = \int_{t \sim \mathbf{w}} s_t \, dt\,, \tag{4}$$

where $t \sim \mathbf{w}$ indicates that t is a parse tree of $\mathbf{w}$ with nonterminal subtypes. Thus the conditional probability density of t given T is

$$P(t|T) = \frac{s_t}{s_T}\,, \tag{5}$$

the conditional probability density of t given $\mathbf{w}$ is

$$P(t|\mathbf{w}) = \frac{s_t}{s_{\mathbf{w}}}\,, \tag{6}$$

and the conditional probability of T given $\mathbf{w}$ is

$$P(T|\mathbf{w}) = \frac{s_T}{s_{\mathbf{w}}} = \int_{t \sim T} P(t|\mathbf{w})\, dt\,. \tag{7}$$

Therefore, we rewrite Equation 1 as

$$\mathcal{L}(\Theta) = -\sum_{i=1}^{m} \log \int_{t \sim T_i} P(t|\mathbf{w}_i)\, dt\,. \tag{8}$$

The derivative of $\mathcal{L}(\Theta)$ with respect to Θ_r, where r is an unrefined production rule, is calculated by Equation 9 in Table 1.

2 Calculation of Analytic Gradients

In GM-LVeGs, Θ_r is the set of parameters in a Gaussian mixture with K_r mixture components:

$$\Theta_r = \{(\rho_{r,k}, \boldsymbol{\mu}_{r,k}, \boldsymbol{\Sigma}_{r,k}) | k = 1, \ldots, K_r\}\,. \tag{13}$$

$$
\begin{aligned}
\mathcal{L}'(\Theta) &= -\sum_{i=1}^{m} \int_{t\sim T_i} \frac{(P(t|\mathbf{w}_i))'}{\int_{t'\sim T_i} P(t'|\mathbf{w}_i)\,dt'}\,dt \\
&= -\sum_{i=1}^{m} \int_{t\sim T_i} \frac{(P(t|\mathbf{w}_i))'}{\int_{t'\sim T_i} P(t'|\mathbf{w}_i)\,dt'} \times \frac{P(t|\mathbf{w}_i)}{P(t|\mathbf{w}_i)}\,dt \\
&= -\sum_{i=1}^{m} \int_{t\sim T_i} \frac{P(t|\mathbf{w}_i)}{\int_{t'\sim T_i} P(t'|\mathbf{w}_i)\,dt'} \times (\log P(t|\mathbf{w}_i))'\,dt \\
&= -\sum_{i=1}^{m} \int_{t\sim T_i} P(t|T_i) \times \left(\log \frac{\prod_{\mathbf{r}\in t} W_r(\mathbf{r})}{\int_{t'\sim \mathbf{w}_i} \prod_{\mathbf{r}\in t'} W_r(\mathbf{r})\,dt'}\right)' dt \\
&= -\sum_{i=1}^{m} \int_{t\sim T_i} P(t|T_i) \times \left(\log \prod_{\mathbf{r}\in t} W_r(\mathbf{r}) - \log \int_{t'\sim \mathbf{w}_i} \prod_{\mathbf{r}\in t'} W_r(\mathbf{r})\,dt'\right)' dt \\
&= -\sum_{i=1}^{m} \left(\int_{t\sim T_i} P(t|T_i) \left(\sum_{\mathbf{r}\in t} \log W_r(\mathbf{r})\right)' dt - \left(\log \int_{t'\sim \mathbf{w}_i} \prod_{\mathbf{r}\in t'} W_r(\mathbf{r})\,dt'\right)'\right) \\
&= -\sum_{i=1}^{m} \left(\int_{t\sim T_i} P(t|T_i) \sum_{\mathbf{r}\in t} \frac{W_r^{'}(\mathbf{r})}{W_r(\mathbf{r})}\,dt - \int_{t'\sim \mathbf{w}_i} P(t'|\mathbf{w}_i) \sum_{\mathbf{r}\in t'} \frac{W_r^{'}(\mathbf{r})}{W_r(\mathbf{r})}\,dt'\right) \\
&= -\sum_{i=1}^{m} \int_{\mathbf{r}} \frac{\mathbb{E}_{P(t|T_i)}[f_r(t)] - \mathbb{E}_{P(t|\mathbf{w}_i)}[f_r(t)]}{W_r(\mathbf{r})} \times W_r^{'}(\mathbf{r})\,d\mathbf{r}\,. \qquad (9)
\end{aligned}
$$

$$
\frac{\partial \mathcal{L}(\Theta)}{\partial \rho_{r,k}} = \sum_{i=1}^{m} \int_{\mathbf{r}} \psi(\mathbf{r}) \cdot \mathcal{N}_{r,k}(\mathbf{r})\,d\mathbf{r}\,. \qquad (10)
$$

$$
\frac{\partial \mathcal{L}(\Theta)}{\partial \boldsymbol{\mu}_{r,k}} = \sum_{i=1}^{m} \int_{\mathbf{r}} \psi(\mathbf{r}) \cdot \mathcal{N}_{r,k}(\mathbf{r}) \cdot \rho_{r,k} \boldsymbol{\Sigma}_{r,k}^{-1}(\mathbf{r} - \boldsymbol{\mu}_{r,k})\,d\mathbf{r}\,. \qquad (11)
$$

$$
\frac{\partial \mathcal{L}(\Theta)}{\partial \boldsymbol{\Sigma}_{r,k}} = \sum_{i=1}^{m} \int_{\mathbf{r}} \psi(\mathbf{r}) \cdot \mathcal{N}_{r,k}(\mathbf{r}) \cdot \rho_{r,k} \boldsymbol{\Sigma}_{r,k}^{-1} \frac{(\mathbf{r} - \boldsymbol{\mu}_{r,k})(\mathbf{r} - \boldsymbol{\mu}_{r,k})^T \boldsymbol{\Sigma}_{r,k}^{-1} - I}{2}\,d\mathbf{r}\,. \qquad (12)
$$

Table 1: Derivation of gradient formulations.

According to Equation 9 in Table 1, we need to take derivatives of $W_r(\mathbf{r})$ with respect to $\rho_{r,k}$, $\boldsymbol{\mu}_{r,k}$, and $\boldsymbol{\Sigma}_{r,k}$ respectively:

$$
\partial W_r(\mathbf{r})/\partial \rho_{r,k} = \mathcal{N}_{r,k}(\mathbf{r})\,, \qquad (14)
$$

$$
\partial W_r(\mathbf{r})/\partial \boldsymbol{\mu}_{r,k} = \rho_{r,k} \mathcal{N}_{r,k}(\mathbf{r}) \boldsymbol{\Sigma}_{r,k}^{-1}(\mathbf{r} - \boldsymbol{\mu}_{r,k})\,, \qquad (15)
$$

$$
\begin{aligned}
\partial W_r(\mathbf{r})/\partial \boldsymbol{\Sigma}_{r,k} = \rho_{r,k} \mathcal{N}_{r,k}(\mathbf{r}) \boldsymbol{\Sigma}_{r,k}^{-1} \frac{1}{2} \Big(&- I \\
&+ (\mathbf{r} - \boldsymbol{\mu}_{r,k})(\mathbf{r} - \boldsymbol{\mu}_{r,k})^T \boldsymbol{\Sigma}_{r,k}^{-1} \Big)\,.
\end{aligned} \qquad (16)
$$

For brevity, we define

$$
\psi(\mathbf{r}) = \frac{\mathbb{E}_{P(t|\mathbf{w}_i)}[f_r(t)] - \mathbb{E}_{P(t|T_i)}[f_r(t)]}{W_r(\mathbf{r})}\,. \qquad (17)
$$

Substituting Equations 14–16 into Equation 9, we have the full gradient formulations of all the parameters (Equations 10–12 in Table 1).

In the following discussion, we assume that all the Gaussians are diagonal. It can be verified that $\psi(\mathbf{r})$ in Equation 17 is in fact a mixture of Gaussians, so multiplying $\psi(\mathbf{r})$ by $\mathcal{N}_{r,k}(\mathbf{r})$ in Equation 10–12 results in another mixture of Gaussians. Below we consider the special case where the resulting Gaussian mixture contains only a single component:

$$
\psi(\mathbf{r}) \cdot \mathcal{N}_{r,k}(\mathbf{r}) = \lambda \cdot \mathcal{N}(\mathbf{r})\,. \qquad (18)
$$

Owing to the sum rule in integral, we can easily extend our derivation on the special case to the general case in which the Gaussian mixture contains multiple components. Because $\mathcal{N}(\mathbf{r})$ is diagonal, it can be factorized as:

$$
\mathcal{N}(\mathbf{r}) = \mathcal{N}(\mathbf{r}^1) \times \cdots \times \mathcal{N}(\mathbf{r}^{|\mathbf{r}|})\,, \qquad (19)
$$

where $\mathcal{N}(\mathbf{r}^1), \ldots, \mathcal{N}(\mathbf{r}^{|\mathbf{r}|})$ are univariate Gaussians (or normal distributions), $\mathbf{r}^d$ $(1 \leq d \leq |\mathbf{r}|)$ refers to the d-th element of $\mathbf{r}$, and $|\mathbf{r}|$ is the dimension of $\mathbf{r}$. The integral in Equation 10 can be readily calculated as

$$\lambda \cdot \int \mathcal{N}(\mathbf{r})\, d\mathbf{r} = \lambda\,. \tag{20}$$

For Equation 11, consider taking the derivative with respect to the mean in dimension d. Since the means in different dimensions are independent, to solve the integral in Equation 11, we only need to solve the following integral:

$$\begin{aligned}
&\lambda \cdot \int \mathcal{N}(\mathbf{r})\mathbf{r}^d\, d\mathbf{r}^1 \ldots d\mathbf{r}^{|\mathbf{r}|} \\
&= \lambda \cdot \int \mathcal{N}(\mathbf{r}^1)\, d\mathbf{r}^1 \times \cdots \times \int \mathcal{N}(\mathbf{r}^d)\mathbf{r}^d\, d\mathbf{r}^d \\
&\times \ldots \times \int \mathcal{N}(\mathbf{r}^{|\mathbf{r}|})\, d\mathbf{r}^{|\mathbf{r}|} \\
&= \lambda \cdot \int \mathcal{N}(\mathbf{r}^d)\mathbf{r}^d\, d\mathbf{r}^d\,.
\end{aligned} \tag{21}$$

For Equation 12, when taking the derivative with respect to the variance in dimension d, we also need to solve Equation 21 and additionally need to solve the following integral:

$$\begin{aligned}
&\lambda \cdot \int \mathcal{N}(\mathbf{r})\mathbf{r}^d\mathbf{r}^d\, d\mathbf{r}^1 \ldots d\mathbf{r}^{|\mathbf{r}|} \\
&= \lambda \cdot \int \mathcal{N}(\mathbf{r}^1)\, d\mathbf{r}^1 \times \cdots \times \int \mathcal{N}(\mathbf{r}^d)\mathbf{r}^d\mathbf{r}^d\, d\mathbf{r}^d \\
&\times \ldots \times \int \mathcal{N}(\mathbf{r}^{|\mathbf{r}|})\, d\mathbf{r}^{|\mathbf{r}|} \\
&= \lambda \cdot \int \mathcal{N}(\mathbf{r}^d)\mathbf{r}^d\mathbf{r}^d\, d\mathbf{r}^d\,.
\end{aligned} \tag{22}$$

The integrals in Equation 21 and Equation 22 are the first order moment and the second order moment of the univariate Gaussian $\mathcal{N}(\mathbf{r}^d)$ respectively, and both of them can be calculated exactly. Therefore, we can calculate analytic gradients of all the parameters in GM-LVeGs.

3 Algorithmic Complexity and Running Time

The time complexity of the learning algorithm for each sentence in each epoch is approximately $\mathcal{O}(cn^3kmd + cklmd^2)$. The first term is the time complexity of the extended inside-outside algorithm, where c is the number of binary productions in CNF, n is the length of the sentence, k is the Gaussian component number of each rule weight function, m is the maximum Gaussian component number of an inside or outside score after pruning, and d is the dimension of diagonal Gaussians. kmd is generally much smaller than cn^3. It shall be noted that m is bounded by k_{max}, which is set to 50 in our experiments. The second term is the approximate time complexity of gradient calculation, where l is the number of times that a production rule is used in all possible parses of a sentence. The second term is much smaller than the first term in general.

We run our learning and inference algorithms with a CPU cluster without any GPU. In our POS tagging experiments, GM-LVeGs are only slightly slower than the baseline LVG models, and we can perform all the tagging experiments on all the datasets with our model within one day. For parsing, there is a trade-off between running time and parsing accuracy based on the amount of pruning. For the best parsing accuracy of GM-LVeGs, it takes two weeks for training. However, once we complete training, parsing can be done within three minutes on the whole testing data of WSJ.

There are a few ways to improve the training efficiency. We currently use CPU parallelization at the sentence level in training, but in the future we may take the advantage of GPU parallelization, e.g., we can vectorize the inside-outside algorithm for a batch of sentences of the same length. Besides, for each long sentence, we can parallelize the inside or outside computation at the same recursive depth.

4 Data Statistics

Dataset	# of tokens	# of sentences		
		train	test	dev
WSJ	950028	39832	2416	1700

Table 2: Statistics of WSJ used for constituency parsing.

Statistics of the data used for constituency parsing are shown in Table 2. Statistics of the data used for POS tagging are summarized in Table 3. In the experiments of constituency parsing, in order to study the influence of the dimension of Gaussians and the number of Gaussian components on the parsing accuracy, we experimented on a small dataset. The small dataset only contains section 4 and section 5 of WSJ. In the two sections, we use file IDs from 80-89 in the two sections are used

	WSJ	English	French	German	Russian	Spanish	Indonesian	Finnish	Italian
# of tokens	1173766	254830	402197	298242	99389	431587	121923	181022	292471
train	38219	12543	14554	14118	4029	14187	4477	12217	12837
dev	5462	2002	1596	799	502	1552	559	716	489
test	5527	2077	298	977	499	274	557	648	489

Table 3: Statistics of WSJ and UD (English, French, German, Russian, Spanish, Indonesian, Finnish, and Italian treebanks). Numbers in the rows of train, test, and dev indicate the number of sentences in training, testing, and development data respectively.

Model	WSJ		English		French		German		Russian		Spanish		Indonesian		Finnish		Italian	
	T	S	T	S	T	S	T	S	T	S	T	S	T	S	T	S	T	S
LVG-D-1	96.50	48.04	91.80	50.79	93.55	30.20	86.52	16.99	81.21	9.24	91.79	22.63	89.08	18.85	83.15	16.82	94.00	37.42
LVG-D-2	96.57	47.60	92.17	52.05	93.86	33.56	86.93	18.32	81.46	10.04	92.10	24.82	89.16	19.21	83.34	18.52	94.45	40.90
LVG-D-4	96.57	48.76	92.30	52.34	93.96	34.90	87.18	19.86	81.95	11.85	92.37	24.82	89.28	19.57	83.76	18.83	94.60	42.54
LVG-D-8	96.60	49.14	92.31	53.06	93.78	34.90	87.52	21.60	81.54	11.25	92.26	23.72	89.23	19.39	83.68	18.67	94.70	42.95
LVG-D-16	96.62	48.74	92.31	52.67	93.75	34.90	87.38	20.98	81.91	12.25	92.47	24.82	89.27	20.29	83.81	19.29	94.81	45.19
LVG-G-1	96.11	43.68	90.84	44.92	92.69	26.51	86.71	17.40	81.22	10.22	91.85	22.63	88.93	18.31	82.94	16.36	93.64	33.74
LVG-G-2	96.27	45.57	92.11	51.37	93.28	28.19	87.87	19.86	81.51	11.45	92.29	23.36	89.19	18.49	83.29	17.44	94.20	38.45
LVG-G-4	96.50	48.19	92.90	54.31	94.06	32.55	88.31	20.78	82.64	11.85	92.58	24.45	89.58	19.03	83.76	19.44	95.00	45.40
LVG-G-8	96.76	50.38	93.29	56.67	94.57	37.25	88.75	21.70	82.85	14.86	92.95	29.20	89.78	20.29	84.69	21.76	95.42	46.83
LVG-G-16	96.78	50.88	93.30	57.54	94.52	34.90	88.92	24.05	84.03	16.63	93.21	27.37	90.09	21.19	85.01	20.53	95.46	48.26
GM-LVeG-D	96.99	53.10	**93.66**	**59.46**	94.73	**39.60**	89.11	24.77	**84.21**	17.84	**93.76**	**32.48**	**90.24**	**21.72**	85.27	**23.30**	95.61	**50.72**
GM-LVeG-S	**97.00**	**53.11**	93.55	58.11	**94.74**	39.26	**89.14**	**25.58**	84.06	**18.44**	93.52	30.66	90.12	**21.72**	**85.35**	22.07	**95.62**	49.69

Table 4: Token accuracy (T) and sentence accuracy (S) for POS tagging on the testing data. The numerical postfix of each LVG model indicates the number of nonterminal subtypes, and hence LVG-G-1 denotes HMM.

for testing, 90-99 for development, and the rest are used for training. The resulting dataset contains 3599 training samples, 426 test samples, and 375 development samples.

5 Additional Experimental Results

The complete experimental results of POS tagging are shown in Table 4. In addition to the results shown in the paper, this table includes the tagging results of LVGs with 1, 2, 4, 8 subtypes for each nonterminal.

In the experiments of constituency parsing, in order to investigate the impact of the maximum Gaussian component number of inside and outside scores, we experiment with a new pruning technique. Specifically, we use a maximum component-pruning threshold k_{hard}. We do not prune any Gaussian component if an inside or outside score has no more than k_{hard} Gaussian components; otherwise we keep only k_{hard} Gaussian components with the largest mixture weights. We experiment on the small dataset mentioned in Section 4. For efficiency, we train GM-LVeG-D only on sentences of no more than 20 words and test GM-LVeG-D only on testing sentences of no more than 25 words. We experiment with $k_{hard} = 10, 20, 30, 40, 50, 60, 70, 80$. The results are shown in Figure 1. We also experiment without component pruning, which corresponds to the rightmost point in Figure 1.

We can see that a weaker component pruning or a larger k_{hard} results in a better F1 score. However, it takes much more time per epoch for learning, as is shown in the lower figure in Figure 1. We find that $k_{hard} = 40$ produces a good F1 score and also admits efficient learning. Therefore, in the experiments of constituency parsing, we use $k_{min} = 40$ in learning for the component-pruning technique introduced in Section 3.2 in the paper.

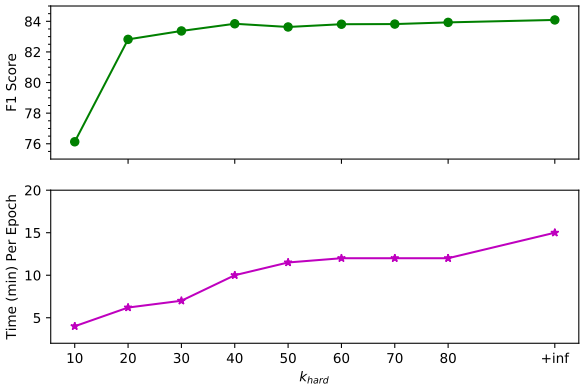

Figure 1: **Upper**: F1 scores with different k_{hard}; **Lower**: time (min) per epoch in learning with different k_{hard}.