Two-Time-Scale Noisy EM Algorithms

Baidu, Cognitive Computing Lab

Belhal Karimi

March 19, 2020

Overview

1. How to Learn in Latent Data Models?

Latent Data Models

- Models where the input-output relationship is not completely characterized by the observed $(x, y) \in \mathsf{X} \times \mathsf{Y}$ pairs in the training set
- Dependence on a set of unobserved latent variables $z \in \mathsf{Z} \subset \mathbb{R}^m$.
- Mandatory: Simulation step to complete the observed data with realizations of the latent variables.
- Formally, this specificity in our setting implies extending the loss function ℓ to accept a third argument as follows:

$$\ell(y, M_{\boldsymbol{\theta}}(x)) = \int_{\mathsf{Z}} \ell(z, y, M_{\boldsymbol{\theta}}(x)) dz \ . \tag{1}$$

Maximizing the Likelihood

- We minimize the following *nonconvex* function on Θ, a convex subset of $\mathbb{R}^d$,

$$\begin{aligned} &\min_{\boldsymbol{\theta} \in \Theta} \overline{\mathcal{L}}(\boldsymbol{\theta}) := \mathrm{R}(\boldsymbol{\theta}) + \mathcal{L}(\boldsymbol{\theta}) \\ &\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_i(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \left\{ -\log g(y_i; \boldsymbol{\theta}) \right\} , \end{aligned} \tag{2}$$

- $\mathrm{R} : \Theta \to \mathbb{R}$ is a smooth convex regularization function.
- $g(y_i; \boldsymbol{\theta})$, is the marginal of the complete data likelihood defined as $f(z_i, y_i; \boldsymbol{\theta})$, i.e.

$$g(y_i; \boldsymbol{\theta}) = \int_{\mathsf{Z}} f(z_i, y_i; \boldsymbol{\theta}) \mu(\mathrm{d}z_i)$$

Exponential Family Setting

- $\{z_i\}_{i=1}^n$ are the (unobserved) latent variables.
- The complete data likelihood belongs to the curved exponential family, *i.e.,*

$$f(z_i, y_i; \boldsymbol{\theta}) = h(z_i, y_i) \exp\left(\left\langle S(z_i, y_i) \,|\, \phi(\boldsymbol{\theta}) \right\rangle - \psi(\boldsymbol{\theta})\right) , \qquad (3)$$

where $\psi(\boldsymbol{\theta})$, $h(z_i, y_i)$ are scalar functions, $\phi(\boldsymbol{\theta}) \in \mathbb{R}^k$ is a vector function, and $S(z_i, y_i) \in \mathbb{R}^k$ is the complete data sufficient statistics.

EM and Variants

- "batch" EM (bEM) method is composed of two steps.
- When $f(z_i, y_i; \boldsymbol{\theta})$ is a curved exponential family model, the E-step amounts to computing the conditional expectation of the complete data sufficient statistics,

$$\bar{\mathbf{s}}(\boldsymbol{\theta}^{(k)}) = \frac{1}{n}\sum_{i=1}^{n} \bar{\mathbf{s}}_i(\boldsymbol{\theta}^{(k)}) \quad \text{where} \quad \bar{\mathbf{s}}_i(\boldsymbol{\theta}) = \int_{Z} S(z_i, y_i) p(z_i|y_i; \boldsymbol{\theta}^{(k)}) \mu(\mathrm{d}z_i)\,. \tag{4}$$

- Then Maximization

$$\text{M-step}: \ \boldsymbol{\theta}^{(k)} = \bar{\theta}(\bar{\mathbf{s}}^{(k)})$$

where $\bar{\mathbf{s}}^{(k)} = \bar{\mathbf{s}}(\boldsymbol{\theta}^{(k)})$

Monte Carlo and Robbins Monro variants

- When expectations (4) are not available (in nonconvex models):
  - Monte Carlo (MC) Approximation:

$$\text{MC-step}: \ \tilde{\mathbf{s}} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{M} \sum_{m=1}^{M} S(z_{i,m}, y_i) \tag{5}$$

    where you draw M samples $z_{i,m} \sim p(z_i | y_i; \theta)$ (direct or MCMC)

- Caveats:
  1. Requires large MC samples M in order to converge.
  2. Do not scale to large n.

2. Two-Time-Scale Approximated EM Algorithms

Large Scale Learning

FIRST LEVEL

- Incremental Updates:

$$\boxed{\text{Incremental-step}: \ \tilde{S}^{(k+1)} = \tilde{S}^{(k)} + \rho_{k+1}\big(\boldsymbol{\mathcal{S}}^{(k+1)} - \tilde{S}^{(k)}\big)}$$

  where $\{\rho_k\}_{k=1}^{\infty} \in [0,1]$ is a sequence of step sizes, $\boldsymbol{\mathcal{S}}^{(k)}$ is a proxy for $\tilde{S}^{(k)}$.

- Several possible updates

$$\text{Incremental} \quad \rho_k = 1 \qquad \boldsymbol{\mathcal{S}}^{(k+1)} = \boldsymbol{\mathcal{S}}^{(k)} + \tfrac{1}{n}\big(\tilde{S}_{i_k}^{(k)} - \tilde{S}_{i_k}^{(\tau_{i_k}^k)}\big)$$

Large Scale Learning

FIRST LEVEL

- Incremental Updates:

$$\boxed{\text{Incremental-step}: \ \tilde{S}^{(k+1)} = \tilde{S}^{(k)} + \rho_{k+1}\big(\boldsymbol{\mathcal{S}}^{(k+1)} - \tilde{S}^{(k)}\big)}$$

  where $\{\rho_k\}_{k=1}^{\infty} \in [0,1]$ is a sequence of step sizes, $\boldsymbol{\mathcal{S}}^{(k)}$ is a proxy for $\tilde{S}^{(k)}$.

- Several possible updates

$$\begin{array}{lll} \text{Incremental} & \rho_k = 1 & \boldsymbol{\mathcal{S}}^{(k+1)} = \boldsymbol{\mathcal{S}}^{(k)} + \frac{1}{n}\big(\tilde{S}_{i_k}^{(k)} - \tilde{S}_{i_k}^{(\tau_{i_k}^k)}\big) \\ \text{Variance Reduction} & \rho_k = cst & \boldsymbol{\mathcal{S}}^{(k+1)} = \tilde{S}^{(\ell(k))} + \big(\tilde{S}_{i_k}^{(k)} - \tilde{S}_{i_k}^{(\ell(k))}\big) \end{array}$$

Large Scale Learning

FIRST LEVEL

- Incremental Updates:

$$\boxed{\text{Incremental-step}: \ \tilde{S}^{(k+1)} = \tilde{S}^{(k)} + \rho_{k+1}\big(\boldsymbol{\mathcal{S}}^{(k+1)} - \tilde{S}^{(k)}\big)}$$

where $\{\rho_k\}_{k=1}^{\infty} \in [0,1]$ is a sequence of step sizes, $\boldsymbol{\mathcal{S}}^{(k)}$ is a proxy for $\tilde{S}^{(k)}$.

- Several possible updates

$$\begin{aligned}
&\text{Incremental} \quad \rho_k = 1 && \boldsymbol{\mathcal{S}}^{(k+1)} = \boldsymbol{\mathcal{S}}^{(k)} + \tfrac{1}{n}\big(\tilde{S}_{i_k}^{(k)} - \tilde{S}_{i_k}^{(\tau_{i_k}^k)}\big) \\
&\text{Variance Reduction} \quad \rho_k = cst && \boldsymbol{\mathcal{S}}^{(k+1)} = \tilde{S}^{(\ell(k))} + \big(\tilde{S}_{i_k}^{(k)} - \tilde{S}_{i_k}^{(\ell(k))}\big) \\
&\text{Fast Incremental} \quad \rho_k = cst && \boldsymbol{\mathcal{S}}^{(k+1)} = \overline{\boldsymbol{\mathcal{S}}}^{(k)} + \big(\tilde{S}_{i_k}^{(k)} - \tilde{S}_{i_k}^{(t_{i_k}^k)}\big) \\
& && \overline{\boldsymbol{\mathcal{S}}}^{(k+1)} = \overline{\boldsymbol{\mathcal{S}}}^{(k)} + n^{-1}\big(\tilde{S}_{j_k}^{(k)} - \tilde{S}_{j_k}^{(t_{j_k}^k)}\big).
\end{aligned}$$

Overcome Large MC Sampling

SECOND LEVEL

- Stochastic Approximation (SA):

$$\boxed{\text{SA-step}: \ \hat{\mathbf{s}}^{(k+1)} = \hat{\mathbf{s}}^{(k)} + \gamma_{k+1}(\tilde{S}^{(k+1)} - \hat{\mathbf{s}}^{(k)})}$$

with decreasing stepsize and $\tilde{S}^{(k+1)}$ MC approximation defined as:

$$\tilde{S}^{(k+1)} = \frac{1}{n}\sum_{i=1}^{n} \tilde{S}_i^{(k+1)} = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{M}\sum_{m=1}^{M} S(z_{i,m}^{(k)}, y_i)$$

- This update converges well with relatively small M. See [Robbins, Monro, 1951] or [Delyon et. al., 1999].
- Then $\boldsymbol{\theta}^{(k+1)} = \bar{\theta}(\hat{\mathbf{s}}^{(k+1)})$

Overcome Large MC Sampling

SECOND LEVEL

- Stochastic Approximation (SA):

$$\boxed{\text{SA-step}: \ \hat{\mathbf{s}}^{(k+1)} = \hat{\mathbf{s}}^{(k)} + \gamma_{k+1}(\tilde{S}^{(k+1)} - \hat{\mathbf{s}}^{(k)})}$$

with decreasing stepsize and $\tilde{S}^{(k+1)}$ MC approximation defined as:

$$\tilde{S}^{(k+1)} = \frac{1}{n}\sum_{i=1}^{n} \tilde{S}_i^{(k+1)} = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{M}\sum_{m=1}^{M} S(z_{i,m}^{(k)}, y_i)$$

- This update converges well with relatively small M. See [Robbins, Monro, 1951] or [Delyon et. al., 1999].
- Then $\boldsymbol{\theta}^{(k+1)} = \bar{\theta}(\hat{\mathbf{s}}^{(k+1)})$

Two-Time-Scale Formulation

Algorithm 1 Two-Time-Scale Noisy EM methods.

1: **Input:** initializations $\hat{\boldsymbol{\theta}}^{(0)} \leftarrow 0$, $\hat{\mathbf{s}}^{(0)} \leftarrow \hat{S}^{(0)}$, $K_{\mathsf{max}} \leftarrow$ max. iteration number.
2: Set the terminating iteration number, $K \in \{0, \ldots, K_{\mathsf{max}} - 1\}$, as a discrete r.v.
3: **for** $k = 0, 1, 2, \ldots, K$ **do**
4: Draw index $i_k \in [\![1, n]\!]$ uniformly (and $j_k \in [\![1, n]\!]$ for fiSAEM).
5: Compute $\hat{S}_i^{(k)}$ using the MC-step , for the drawn indices.
6: Compute the surrogate sufficient statistics $\boldsymbol{\mathcal{S}}^{(k+1)}$ using different updates.
7: First Level: Compute $\hat{S}^{(k+1)}$ via the Incremental-step.
8: Second Level:Compute $\hat{\mathbf{s}}^{(k+1)}$ via the SA-step .
9: Compute $\hat{\boldsymbol{\theta}}^{(k+1)}$ via the M-step.
10: **end for**
11: **Return**: $\hat{\boldsymbol{\theta}}^{(K)}$.

Intuition Behind The Two Stages

- First Level: Incremental and Variance Reduction
  - **Incremental** updates to scale to large datasets. See [Neal and Hinton, 1998], [Bottou and Bousquet, 2008].
  - **Variance reduction** to control variance induced by incremental sampling. See [Johnson et. al., 2013], [Karimi et. al., 2019].
- Second Level: Robbins Monro update/ Pointwise convergence
  - Robbins Monro update. Decreasing stepsize to smooth the iterates.
  - Smaller Monte Carlo batchsize M.
  - Kind of like averaging scheme (memory term in the drift term). See [Polyak, Ruppert, 1990].

Intuition: Variance Reduction

- Need to temper the variance induced by **incremental** sampling.
- See SVRG [Johnson et. al., 2013] or SAGA [Defazio et. al., 2014] in optimization literature.
- The whole point is to temper the variance term

  $$\mathbb{E}[\|\hat{\boldsymbol{s}}^{(k)} - \boldsymbol{\mathcal{S}}^{(k+1)}\|^2]$$

  Depending on the update, this term can be controlled to increase speed of convergence.
- **Control variate**, as we are using it here, can be used for other algorithms. See control variate for MCMC [Brosse et. al., 2019].

Intuition: Control MC Fluctuations

- Recall: expectations are never available and requires Monte Carlo approximation.
- There are errors (MC fluctuations) when approximating the expectation $\bar{\mathbf{s}}_i(\hat{\boldsymbol{\theta}}(\hat{\mathbf{s}}^{(k-1)}))$.

$$\eta_{i,\vartheta} := \frac{1}{\sqrt{M}} \sum_{m=1}^{M} \left\{ \tilde{S}_i(y_i, z_{i,m}) - \bar{\mathbf{s}}_i(\vartheta) \right\} \tag{6}$$

- We want and need to control the $\sup_{\vartheta \in \Theta}$ of this quantity
- Standard assumption in empirical processes and stochastic optimization
- Have recourse to Dudley's inequality and Bracketing Number
- BUT **curse of dimensionality**

Intuition: Control MC Fluctuations

- In [Vershynin, High-Dimensional Probability, 2018]:

$$\mathbb{E} \sup_{f \in \mathcal{F}} \left| \frac{1}{M} \sum_{i=1}^{M} f(X_i) - \mathbb{E}\left[f(X) \right] \right| \leq \frac{CL}{\sqrt{M}}$$

- In [Wainwright, High-Dimensional Statistics, 2019], the application of the Dudley's inequality yields:

$$\mathbb{E} \sup_{f} |X_f| = \mathbb{E} \sup_{f \in \mathcal{F}} |X_f - X_0| \leq \frac{1}{\sqrt{M}} \int_0^1 \sqrt{\log \mathcal{N}(\mathcal{F}, \|\cdot\|_\infty, \varepsilon)} d\varepsilon$$

  where $\mathcal{N}(\mathcal{F}, \|\cdot\|_\infty, \varepsilon)$ is the bracketing number and ϵ denotes the level of approximation (the bracketing number goes to infinity when $\epsilon \to 0$)

- In [Van Der Vaart, Asymptotic Statistics, 2000]:

$$\mathcal{N}(\mathcal{F}, \|\cdot\|_\infty, \varepsilon) \leq K \left(\frac{\operatorname{diam} \Theta}{\varepsilon} \right)^d, \quad \text{every} \quad 0 < \varepsilon < \operatorname{diam} \Theta$$

Finite-Time Analysis

To set our stage, we consider the minimization problem:

$$\min_{\mathbf{s}\in\mathsf{S}} \; V(\mathbf{s}) := \overline{\mathcal{L}}(\overline{\boldsymbol{\theta}}(\mathbf{s})) = \mathrm{R}(\overline{\boldsymbol{\theta}}(\mathbf{s})) + \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_i(\overline{\boldsymbol{\theta}}(\mathbf{s})), \tag{7}$$

where $\overline{\boldsymbol{\theta}}(\mathbf{s})$ is the unique map defined in the M-step (**??**).

Lemma 1

Assume (A1) to (A4). For all $\mathbf{s}, \mathbf{s}' \in \mathsf{S}$ *and* $i \in [\![1, n]\!]$, *we have*

$$\|\overline{\mathbf{s}}_i(\overline{\boldsymbol{\theta}}(\mathbf{s})) - \overline{\mathbf{s}}_i(\overline{\boldsymbol{\theta}}(\mathbf{s}'))\| \le \mathsf{L}_{\mathbf{s}} \, \|\mathbf{s} - \mathbf{s}'\|, \;\; \|\nabla V(\mathbf{s}) - \nabla V(\mathbf{s}')\| \le \mathsf{L}_V \, \|\mathbf{s} - \mathbf{s}'\|, \tag{8}$$

where $\mathsf{L}_{\mathbf{s}} := C_{\mathsf{Z}} \, \mathsf{L}_p \, \mathsf{L}_\theta$ *and* $\mathsf{L}_V := \upsilon_{\max}\big(1 + \mathsf{L}_{\mathbf{s}}\big) + \mathsf{L}_B \, C_{\mathsf{S}}$.

Finite-Time Analysis

Theorem 1

- *Consider the iSAEM method. There exists a universal constant $\mu \in (0,1)$ (independent of n) such that if we set the step size as $\gamma_k \propto 1/k^\alpha$.*

$$\begin{aligned}
&\sum_{k=0}^{K_{\max}-1} \alpha_k \mathbb{E}\left[\left\| \bar{s} \circ \mathsf{T}\left(\widehat{S}^k\right) - \widehat{S}^k \right\|^2\right] \\
&\leq n \frac{2\overline{L}_{\mathsf{v}}}{\mu K_{\max}} \frac{\upsilon_{\max}^2}{\upsilon_{\min}^2} \mathbb{E}[V(\hat{\boldsymbol{s}}^{(0)}) - V(\hat{\boldsymbol{s}}^{(K_{\max})})] \\
&+ O\big(\sum_{k=1}^{K_{\max}} \sum_{i=1}^{n} \eta_{i,\theta^{(\tau_i^k)}}^{(k)}\big)
\end{aligned} \tag{9}$$

- *Similar to linear rate of incremental EM (deterministic) PLUS a Monte Carlo noise term.*

Finite-Time Analysis

- Also we can show:

$$\frac{1}{v_{\max}^2} \sum_{k=0}^{K_{\max}-1} \alpha_k \mathbb{E}\left[\left\|\dot{V}\left(\widehat{S}^k\right)\right\|^2\right] \leq \sum_{k=0}^{K_{\max}-1} \alpha_k \mathbb{E}\left[\left\|\bar{s} \circ \top\left(\widehat{S}^k\right) - \widehat{S}^k\right\|^2\right] \tag{10}$$

- which gives a bound on the gradient of the Lyapunov function V.

3. Numerical Experiments

Gaussian Mixture Models

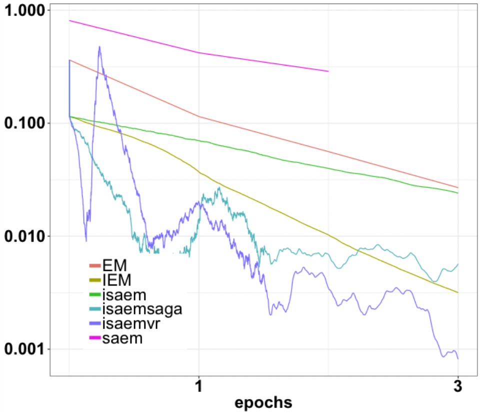
- Fit a GMM model to a set of n observations $\{y_i\}_{i=1}^n$ whose distribution is modeled as a Gaussian mixture of M components, each with a unit variance.
- $z_i \in [\![M]\!]$ are the latent labels, the complete log-likelihood is:

$$\begin{aligned}\log f(z_i, y_i; \boldsymbol{\theta}) = &\sum_{m=1}^{M} 1_m(z_i)\left[\log(\omega_m) - \mu_m^2/2\right] \\ &+ \sum_{m=1}^{M} 1_m(z_i)\mu_m y_i + \text{constant}\ .\end{aligned} \tag{11}$$

  where $\boldsymbol{\theta} := (\boldsymbol{\omega}, \boldsymbol{\mu})$ with $\boldsymbol{\omega} = \{\omega_m\}_{m=1}^{M-1}$ are the mixing weights with t $\omega_M = 1 - \sum_{m=1}^{M-1} \omega_m$ and $\boldsymbol{\mu} = \{\mu_m\}_{m=1}^{M}$ are the means.
- We use the penalization $\mathrm{R}(\boldsymbol{\theta}) = \frac{\delta}{2}\sum_{m=1}^{M} \mu_m^2 - \log \mathrm{Dir}(\boldsymbol{\omega}; M, \epsilon)$ where $\delta > 0$ and $\mathrm{Dir}(\cdot; M, \epsilon)$ is the M dimensional symmetric Dirichlet distribution with concentration parameter $\epsilon > 0$.
- Generate samples from a GMM model with $M = 2$, $\mu_1 = -\mu_2 = 0.5$.

Gaussian Mixture Models

Fixed sample size We use $n = 10^4$ synthetic samples and run to get $\mu^\star$. We compare the bEM, iEM, iSAEM, vrSAEM and fiSAEM methods. RM stepsize is $\gamma_k = 1/k^{0.6}$, and for vrSAEM and fiSAEM, ρ_k is constant and proportional to $1/n^{2/3}$. We average over 5 independent runs for each method using the same stepsizes as in the finite sample size case above.



Deformable Template for Image Analysis

- $(y_i, i \in [\![1, n]\!])$ images modelled as deformation of a template.
- The model reads as follows:

$$y_i(s) = I\left(x_s - \Phi_i\left(x_s\right)\right) + \sigma\varepsilon_i(s)$$

  where s is the pixel index, x_s its coordinate, I the template and Φ_i the deformation.

- The template model given p_k landmarks on template and a fixed kernel:

$$I_\xi = \mathbf{K_p}\xi, \quad \text{where} \quad \left(\mathbf{K_p}\xi\right)(x) = \sum_{k=1}^{k_p} \mathbf{K_p}\left(x, p_k\right)\xi(k) \tag{12}$$

- The deformation model given landmarks and a fixed kernel:

$$\Phi_i(x) = \left(\mathbf{K_g} z_i\right)(x) = \sum_{k=1}^{k_s} \mathbf{K_g}\left(x, g_k\right)\left(z_i^{(1)}(k), z_i^{(2)}(k)\right) \tag{13}$$

- We learn the parameters $\boldsymbol{\theta} = (\sigma, \xi, \Gamma)$ using the two-time-scale methods.

Deformable Template for Image Analysis

- USPS Digits Dataset
- For a credit of epochs (running time maybe?), we generate images using the learnt model and see which one are similar to the template.

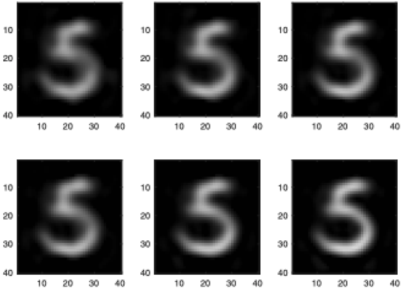


Figure 2: Estimation of the template: first row : using SAEM (benchmark) ; second row : using new incremental method with minibatch size of 0.1; columns correspond to 1, 2 and 3 epochs, respectively.

Deformable Template for Image Analysis

- For a credit of number of images used in training.

Figure 3: Synthetic images sampled from the model for digit 5 using the parameter estimates obtained with the batch version on 20 images (top) and with the mini-batch version with 1/5th of the data with 100 images.

ONGOING TASKS

- Implement Deformable Template analysis on USPS digits
- Proofs for Variance Reduction and Fast Incremental Two-Time-Scale methods
- Finish writing

Thank you! Questions?