A Bayesian Framework for Online Classifier Ensemble

Qinxun Bai
Computer Science
Boston University

Prof. Henry Lam
Industrial & Operations
University of Michigan

Prof. Stan Sclaroff
Computer Science
Boston University

Ensemble Classifier Learning

Idea

- Learn a set of classifiers from a fixed training set
- Combine them together to form an ensemble classifier, most commonly linear combination

Representative methods

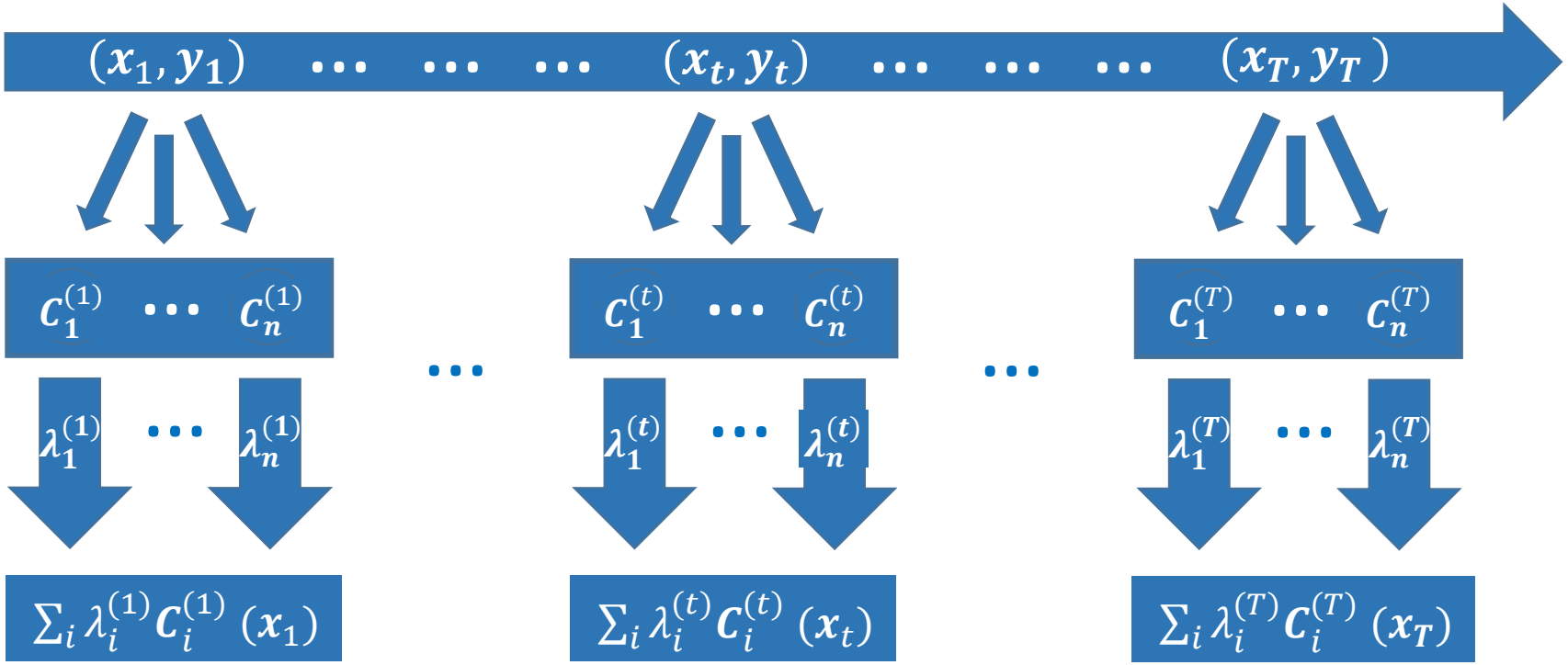
- bagging, boosting, random forest

Why it helps?

- bagging: reduce the variance by averaging
- boosting: several perspectives, but not completely understood yet

Online Ensemble Learning

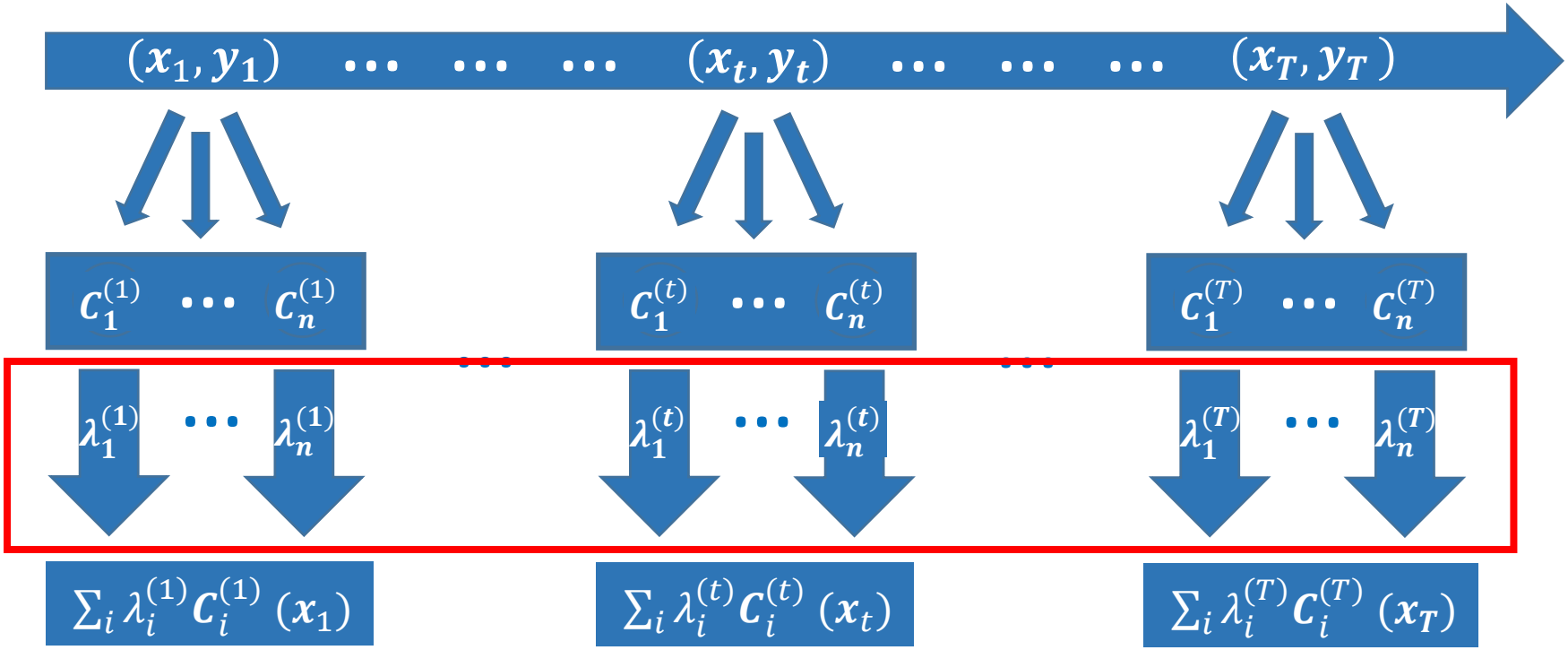
Training sample $(\boldsymbol{x}_t, y_t)$ comes in sequentially, update the model by processing each training sample once "on arrival"



Online Ensemble Learning

Our focus

- Given weak classifiers $(C_1^{(t)}, C_2^{(t)}, \ldots, C_n^{(t)})$ at each time step, find the optimal weight vector $(\lambda_1^{(t)}, \lambda_2^{(t)}, \ldots, \lambda_n^{(t)})$



Loss Minimization Setup

$$\lambda^* = argmin_{\boldsymbol{\lambda}} E_{p(\boldsymbol{x},y)} l(\boldsymbol{\lambda}; \boldsymbol{x}, y)$$

$p(\boldsymbol{x}, y)$ unknown, solve for $min_{\boldsymbol{\lambda}} \sum_{\boldsymbol{t=1}}^{\boldsymbol{T}} l\left(\boldsymbol{\lambda}; \boldsymbol{x}^{(t)}, y^{(t)}\right)$

- $g_i^{(t)} := g\left(C_i\left(\boldsymbol{x}^{(t)}\right), y^{(t)}\right)$: individual loss of C_i at step t
- $\boldsymbol{g}^{(t)} = \left(g_1^{(t)}, g_2^{(t)}, \ldots, g_n^{(t)}\right)$: weak classifier losses at step t
- $l\left(\boldsymbol{\lambda}; \boldsymbol{g}^{(t)}\right)$: ensembled loss w.r.t. $\boldsymbol{\lambda}$ at step t, e.g. $\sum_i \lambda_i \, g_i^{(t)}$
- $L_T(\boldsymbol{\lambda}; \boldsymbol{g}^{(1:T)}) = l_0(\boldsymbol{\lambda}) + \sum_{t=1}^{T} l\left(\boldsymbol{\lambda}; \boldsymbol{g}^{(t)}\right)$: cumulative loss up to T

Main Results

A Bayesian scheme for online classifier ensemble

- Formulate as a stochastic optimization problem and estimate the ensemble weights through a recursive Bayesian procedure
- Under some regularity conditions, converges to global optimum, in contrast with many local methods
- Convergence rate superior to standard stochastic gradient descent, in terms of asymptotic variance
- Promising performance in real-world data experiments

A Recursive Bayesian Scheme

Two classical results in Bayesian statistics

- The Bayesian posterior distribution tends to peak at the MLE of the same likelihood function (Chen 1985)
- MLE minimizes the expected negative log-likelihood

A Recursive Bayesian Scheme

Two classical results in Bayesian statistics

- The Bayesian posterior distribution tends to peak at the MLE of the same likelihood function (Chen 1985)
- MLE minimizes the expected negative log-likelihood

Our idea

- Above results hold regardless of whether the likelihood actually describes the data generating process or not
- Derive an artificial likelihood from a predefined loss function and run the recursive Bayesian procedure

Algorithm

To solve: $min_{\boldsymbol{\lambda}}\, l_0(\boldsymbol{\lambda}) + \sum_{t=1}^{T} l\left(\boldsymbol{\lambda}; \boldsymbol{g}^{(t)}\right)$

- prior: $p_0(\boldsymbol{\lambda}) = e^{-l_0(\boldsymbol{\lambda})}$
- likelihood: $p_l(\boldsymbol{g}|\boldsymbol{\lambda}) = e^{-l(\boldsymbol{\lambda};\boldsymbol{g})}$

Algorithm

To solve: $min_{\boldsymbol{\lambda}}\, l_0(\boldsymbol{\lambda}) + \sum_{t=1}^{T} l\left(\boldsymbol{\lambda}; \boldsymbol{g}^{(t)}\right)$

- prior: $p_0(\boldsymbol{\lambda}) = e^{-l_0(\boldsymbol{\lambda})}$
- likelihood: $p_l(\boldsymbol{g}|\boldsymbol{\lambda}) = e^{-l(\boldsymbol{\lambda};\boldsymbol{g})}$

for $t = 1$ to T do

- For sample $(\boldsymbol{x}_t, y_t)$, compute $g_i^{(t)}$ for all weak classifiers
- Update the "posterior distribution" of $\boldsymbol{\lambda}$

$$p(\boldsymbol{\lambda}|\boldsymbol{g}^{(1:t)}) \propto p_l\left(\boldsymbol{g}^{(t)}|\boldsymbol{\lambda}\right) \cdot p(\boldsymbol{\lambda}|\boldsymbol{g}^{(1:t-1)})$$

- Update weak classifiers using $(\boldsymbol{x}_t, y_t)$

A Specific Example

Choice of loss

$$l(\boldsymbol{\lambda}; \boldsymbol{g}) = \theta \sum_{i=1}^{m} \lambda_i g_i - \sum_{i=1}^{m} \log \lambda_i$$

- $\sum_{i=1}^{m} \lambda_i g_i$: weighted sum of individual loss
- $\sum_{i=1}^{m} \log \lambda_i$: regularizer, prevents the trivial minimizer λ_i=0 for all i
- θ: trade-off parameter

A Specific Example

Exponential likelihood

$$p_l(\boldsymbol{g}|\boldsymbol{\lambda}) = \prod_{i=1}^{m} (\theta\lambda_i) e^{-\theta\lambda_i g_i}$$

Gamma prior

$$p(\boldsymbol{\lambda}) \propto \prod_{i=1}^{m} \lambda_i^{\alpha-1}\, e^{-\beta\lambda_i}$$

Conjugacy the posterior is again Gamma

$$p(\boldsymbol{\lambda}|\boldsymbol{g}^{(1:t)}) \propto \prod_{i=1}^{m} \lambda_i^{\alpha+t-1}\, e^{-(\beta+\theta\sum_{s=1}^{t} g_i^s)\lambda_i}$$

A Specific Example

Posterior mean for each λ_i

$$\bar{\lambda}_i = \frac{\alpha + t}{\beta + \theta \sum_{s=1}^{t} g_i^s}$$

Ensemble loss based prediction rule

$$y = \begin{cases} 1 & if \sum_{i=1}^{m} \bar{\lambda}_i g_i(x, 1) \leq \sum_{i=1}^{m} \bar{\lambda}_i g_i(x, -1) \\ -1 & otherwise \end{cases}$$

We derived an error bound for this prediction rule

Theoretical Guarantees

Global Convergence

- The posterior distribution of $\boldsymbol{\lambda}$ converges to the cumulative loss minimizer $\boldsymbol{\lambda}_T^*$ under asymptotic normality
- The posterior mean provides a tight approximation to $\boldsymbol{\lambda}_T^*$
- $\boldsymbol{\lambda}_T^*$ is the global optimum (no convexity is required)!
- Under some regularity conditions on the loss function

Theoretical Guarantees

Global Convergence

- The posterior distribution of $\boldsymbol{\lambda}$ converges to the cumulative loss minimizer $\boldsymbol{\lambda}_T^*$ under asymptotic normality
- The posterior mean provides a tight approximation to $\boldsymbol{\lambda}_T^*$
- $\boldsymbol{\lambda}_T^*$ is the global optimum (no convexity is required)!

Convergence Rate

- Our method converges strictly faster than standard SGD, except when the step size and Hessian matrix are chosen optimally for SGD.

Real World Experiments

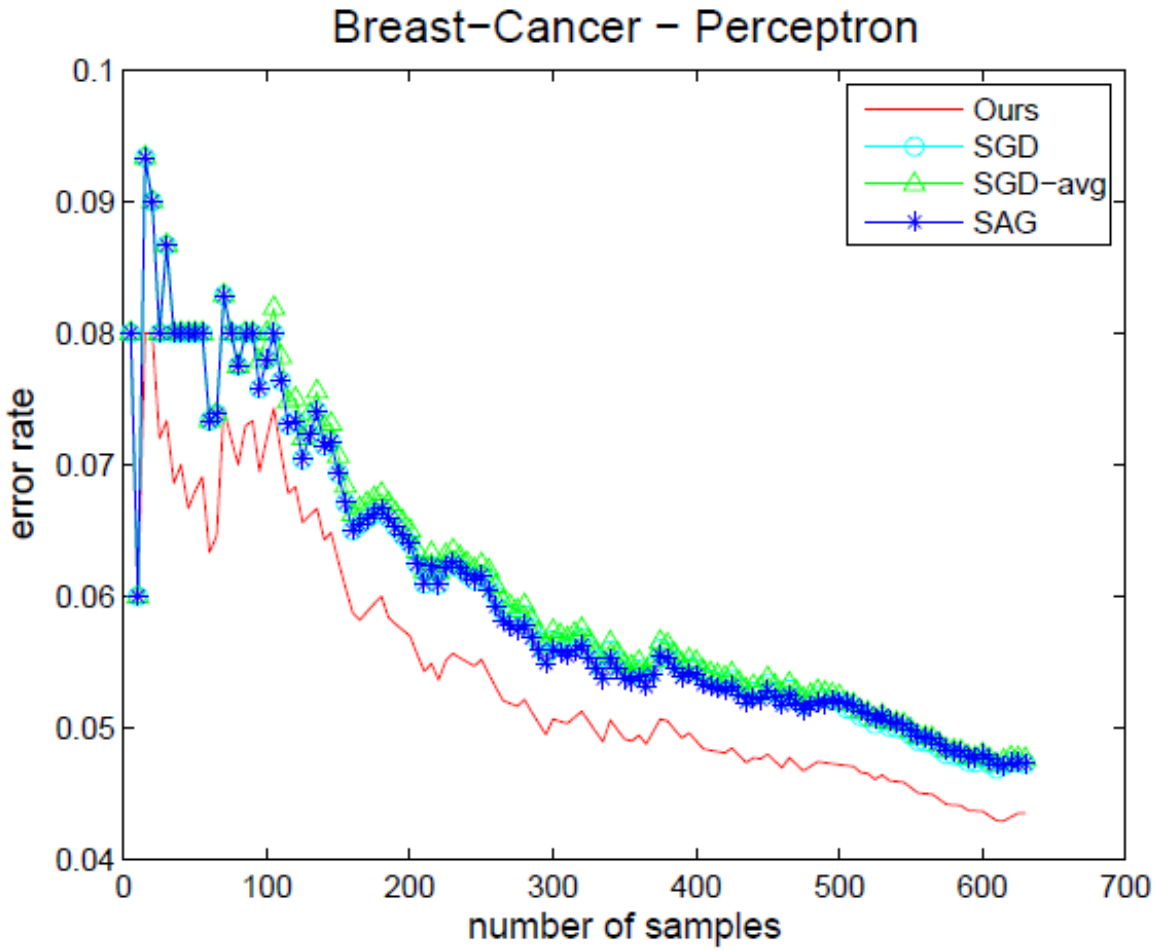
Our method vs. three SGD baselines

- Standard SGD, Polyak Averaging and Stochastic Averaging
- Consistently outperforms all three baselines

Our method vs. online boosting algorithms

- Compare with three representative online boosting algorithms
- Often compares favorably

Error Rate Behavior vs. Baselines



SGD: Standard SGD
SGD-avg: Polyak Averaging
SAG: Stochastic Averaging

Summary

A Bayesian scheme for online classifier ensemble

- Straightforward and easy to implement
- For a specified class of loss functions, possesses strong theoretical guarantees:
  - global convergence
  - superior convergence rate compared with standard SGD
- Promising performance in practice

Likelihood interpretation

A likelihood function derived in this way is not necessarily a proper distribution, our theorems work for either cases. The convergence analysis relies on the Laplace method, which is non-probabilistic in nature.

Not all likelihood function allows a closed form Bayesian update, but for the linear ensemble loss function in our problem, we obtained closed form update by choosing a conjugate prior.

Why superior convergence rate

- The error between our estimate of the weights and the true optimum can be decomposed as: error between MLE and true optimum + error between Bayesian posterior mean and MLE. The fact is that MLE enjoys superior asymptotic property (so-called efficient estimator, in the sense that its asymptotic variance, which controls the convergence rate, is the best possible). And we show that the error between Bayesian posterior mean and MLE is of order less than the first error component. Hence the result.

Conditions for $L_T(\boldsymbol{\lambda}; \boldsymbol{g}^{1:T})$

Regularity conditions

- "local optimality": $\nabla L_T\left(\boldsymbol{\lambda}_T^*; \boldsymbol{g}^{(1:T)}\right) = 0$ and $\nabla^2 L_T\left(\boldsymbol{\lambda}_T^*; \boldsymbol{g}^{(1:T)}\right)$ is positive definite
- "steepness": minimum eigenvalue of $\nabla^2 L_T\left(\boldsymbol{\lambda}_T^*; \boldsymbol{g}^{(1:T)}\right)$ diverges to ∞
- "smoothness": continuity of the $\nabla^2 L_T\left(\boldsymbol{\lambda}_T^*; \boldsymbol{g}^{(1:T)}\right)$
- "concentration": tail of the $L_T\left(\boldsymbol{\lambda}_T^*; \boldsymbol{g}^{(1:T)}\right)$ can be ignored asymptotically
- "Integrability": $\int e^{-L_T(\boldsymbol{\lambda}; \boldsymbol{g}^{(1:T)})} d\boldsymbol{\lambda} < \infty$