Machine Learning and Neural Networks (MATH3431) Epiphany term, 2025

Lecture notes 5: Stochastic gradient descent

Lecturer & author: Georgios P. Karagiannis georgios.karagiannis@durham.ac.uk

Aim. To introduce the stochastic gradient descent (motivation, description, practical tricks, analysis in the convex scenario, and implementation).

Reading list & references:

(1) Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.
- Ch. 14.3 Stochastic Gradient Descent (SGD), 14.5 Variants, 14.5 Learning with SGD

(2) Bottou, L. (2012). Stochastic gradient descent tricks. In Neural networks: Tricks of the trade (pp. 421-436). Springer, Berlin, Heidelberg.

1. Motivations for stochastic gradient descent

Problem 1. Consider a learning problem $(\mathcal{H}, \mathcal{Z}, \ell)$. Learning may involve the computation of the minimizer $w^* \in \mathcal{W}$ of the risk function (RF) $R_g(w) = \mathrm{E}_{z\sim g}(\ell(h_w, z))$ given an unknown data generating model $g(\cdot)$, using a known tractable loss $\ell(\cdot,\cdot)$, and hypothesis $h_w \in \mathcal{H}$; that is

$$w^* = \arg\min_w (R_g(w)) = \arg\min_w (\mathrm{E}_{z\sim g}(\ell(h_w, z))) \tag{1.1}$$

Note 2. Gradient descent (GD) cannot be directly utilized to address Problem 1 (i.e., minimize the Risk function) because g is unknown, and because (1.1) may involve a computationally intractable integration. Instead GD aims to minimize the ERF $\hat{R}(w) = \frac{1}{n}\sum_{i=1}^n \ell(h_w, z_i)$ which ideally can be considered as a proxy when data size n is big (big-data).

Note 3. The implementation of GD may be computationally impractical even in problems aiming to minimize ERF $\hat{R}_n(w)$ if we have big data ($n \approx$big). This is because GD requires the recursive computation of the exact gradient $\partial_w \hat{R}_n(w) = \left\{\frac{1}{n}\sum_{i=1}^n v_i : v_i \in \partial_w \ell(h_w, z_i)\right\}$ using (required to scan through) all the data $\{z_i\}$ at each iteration. That may be too slow.

Note 4. Stochastic gradient descent (SGD) aims at solving (1.1), and overcoming the issues in Notes 2 & 3 by using an unbiased estimator of the actual gradient (or some sub-gradient) based on a sample (a single example or a set of examples) properly drawn from g.

2. Stochastic gradient descent

Problem 5. For the sake of notation simplicity and generalization, we present Stochastic Gradient Descent (SGD) in the following minimization problem

$$w^* = \arg\min_w (f(w)) \tag{2.1}$$

where here $f : \mathbb{R}^d \to \mathbb{R}$, and $w \in \mathcal{W} \subseteq \mathbb{R}^d$; $f(\cdot)$ is the unknown function to be minimized, e.g., $f(\cdot)$ can be the risk function $R_g(w) = \mathrm{E}_{z\sim g}(\ell(h_w, z))$.

Algorithm 6. *Stochastic Gradient Descent (SGD) with learning rate $\eta_t > 0$ for Problem 5*

For $t = 1, 2, 3, ...$ iterate:

(1) *compute*

$$w^{(t+1)} = w^{(t)} - \eta_t v_t, \tag{2.2}$$

where v_t is a random vector such that $E\left(v_t | w^{(t)}\right) \in \partial f\left(w^{(t)}\right)$

(2) *terminate if a termination criterion is satisfied, e.g.*

$$\textit{If } t \geq T_{\max} \textit{ then STOP}$$

Note 7. If f is differentiable at $w^{(t)}$, it is $\partial f\left(w^{(t)}\right) = \left\{\nabla f\left(w^{(t)}\right)\right\}$. Hence v_t is such as $\mathrm{E}\left(v_t | w^{(t)}\right) = \nabla f\left(w^{(t)}\right)$ in Algorithm 6 step 1.

Note 8. Assume f is differentiable (for simplicity). To compare SGD with GD, we can re-write (2.2) in the SGD Algorithm 6 as

$$w^{(t+1)} = w^{(t)} - \eta_t \left[\nabla f\left(w^{(t)}\right) + \xi_t\right], \tag{2.3}$$

where

$$\xi_t := v_t - \nabla f\left(w^{(t)}\right)$$

represents the (observed) noise introduced in (2.2) by using a random realization of the exact gradient.

Note 9. Given T SGD algorithm iterations, the output of SGD can be (but not a exclusively)

(1) the average (after discarding the first few iterations of $w^{(t)}$ for stability reasons)

$$w_{\mathrm{SGD}}^{(T)} = \frac{1}{T}\sum_{t=1}^{T} w^{(t)} \tag{2.4}$$

(2) or the best value discovered

$$w_{\mathrm{SGD}}^{(T)} = \arg\min_{\{w_t\}}\left(f\left(w^{(t)}\right)\right)$$

(3) or the last value discovered

$$w_{\mathrm{SGD}}^{(T)} = w^{(T)}$$

Note 10. SGD output converges to a local minimum, $w_{\mathrm{SGD}}^{(T)} \to w_*$ (in some sense), under different sets of regularity conditions –Section 3 has a brief analysis. To achieve this, Conditions 11 on the learning rate are rather inevitable and should be satisfied.

Condition 11. Regarding the learning rate (or gain) $\{\eta_t\}$ should satisfy conditions

(1) $\eta_t \geq 0$,

(2) $\sum_{t=1}^{\infty}\eta_t=\infty$
(3) $\sum_{t=1}^{\infty}\eta_t^2<\infty$

Note 12. The popular learning rates $\{\eta_t\}$ in Note 14 in Lect. notes 4 "Gradient descent" satisfy Condition 11 and hence can be used in SGD too. Once parameterized, η_t can be tuned based on pilot runs using a reasonably small fraction of the training data set.

Note 13. (Intuition on Condition 11). Assume that v_t is bounded. Condition 11(3) aims at reducing the effect of the randomness in v_t (introduced noise ξ_t) because it implies $\eta_t \searrow 0$ as $t\to\infty$; if this was not the case then

$$w^{(t+1)}-w^{(t)}=-\eta_t v_t\to 0$$

may not be satisfied and the chain $\left\{w^{(t)}\right\}$ may not converge. Condition 11(2) prevents η_t from reducing too fast and allows the generated chain $\left\{w^{(t)}\right\}$ to be able to converge. E.g., after t iterations

$$\begin{aligned}\left\|w^{(t)}-w^*\right\|&=\left\|w^{(t)}\pm w^{(0)}-w^*\right\|\geq\left\|w^{(0)}-w^*\right\|-\left\|w^{(t)}-w^{(0)}\right\|\\&\geq\left\|w^{(0)}-w^*\right\|-\sum_{t=0}^{\infty}\left\|w^{(t+1)}-w^{(t)}\right\|=\left\|w^{(0)}-w^*\right\|-\sum_{t=0}^{T-1}\left\|\eta_t v_t\right\|\end{aligned}$$

However if it was $\sum_{t=1}^{\infty}\eta_t<\infty$ it would be $\sum_{t=0}^{\infty}\|\eta_t v_t\|<\infty$ and hence $w^{(t)}$ would never converge to w^* if the seed $w^{(0)}$ is far enough from w^*.

3. Analysis of SGD (Algorithm 6)

Note 14. Recall (Note 8) that the stochasticity of SGD comes from the stochastic sub-gradients $\{v_t\}$ in (2.2); hence the expectations below are under these random vectors' distributions.

Proposition 15. *Let $f(\cdot)$ be a convex function. If we run SGD algorithm of f with learning rate $\eta_t>0$ for T steps, the output $w_{\text{SGD}}^{(T)}=\frac{1}{T}\sum_{t=1}^{T}w^{(t)}$ satisfies*

$$\mathrm{E}\left(f\left(w_{\text{SGD}}^{(T)}\right)\right)-f(w^*)\leq\frac{\|w^*\|^2}{2\eta T}+\frac{\eta}{2}\frac{1}{T}\sum_{t=1}^{T}\mathrm{E}\|v_t\|^2 \tag{3.1}$$

Proof. Let $v_{1:t}=(v_1,...,v_t)$. By Jensens' inequality (or see 4.3 in Lect. notes 4)

$$\mathrm{E}\left(f\left(w_{\text{SGD}}^{(T)}\right)-f(w^*)\right)\leq\mathrm{E}\left(\frac{1}{T}\sum_{t=1}^{T}\left(f\left(w^{(t)}\right)-f(w^*)\right)\right)=\frac{1}{T}\sum_{t=1}^{T}\mathrm{E}\left(f\left(w^{(t)}\right)-f(w^*)\right) \tag{3.2}$$

I will try to use Lemma 17 from Lect. notes 4, hence I need to show

$$\mathrm{E}\left(f\left(w^{(t)}\right)-f(w^*)\right)\leq\mathrm{E}\left(\langle w^{(t)}-w^*,v_t\rangle\right) \tag{3.3}$$

where the expectation is under $v_{1:T}$. It is

$$\begin{aligned}\mathrm{E}_{v_{1:T}}\left(\langle w^{(t)}-w^*,v_t\rangle\right)=&\mathrm{E}_{v_{1:t}}\left(\langle w^{(t)}-w^*,v_t\rangle\right)\\=&\mathrm{E}_{v_{1:t-1}}\left(\mathrm{E}_{v_{1:t}}\left(\langle w^{(t)}-w^*,v_t\rangle|v_{1:t-1}\right)\right)\qquad(\text{ law of total expectation })\end{aligned}$$

But $w^{(t)}$ is fully determined by $v_{1:t-1}$, (see (2.2)) so

$$\mathrm{E}_{v_{1:t-1}}\left(\mathrm{E}_{v_{1:t}}\left(\langle w^{(t)}-w^{*},v_{t}\rangle|v_{1:t-1}\right)\right)=\mathrm{E}_{v_{1:t-1}}\left(\langle w^{(t)}-w^{*},\mathrm{E}_{v_{1:t}}\left(v_{t}|v_{1:t-1}\right)\rangle\right)$$

As $w^{(t)}$ is fully determined by $v_{1:t-1}$ then $\mathrm{E}_{v_{1:t}}\left(v_{t}|v_{1:t-1}\right)=\mathrm{E}_{v_{1:t}}\left(v_{t}|w^{(t)}\right)\in\partial f\left(w^{(t)}\right)$, hence $\mathrm{E}_{v_{1:t}}\left(v_{t}|v_{1:t-1}\right)$ is a sub-gradient. By sub-gradient definition

$$\begin{aligned}\mathrm{E}_{v_{1:t-1}}\left(\langle w^{(t)}-w^{*},\mathrm{E}_{v_{1:t}}\left(v_{t}|v_{1:t-1}\right)\rangle\right)\geq&\mathrm{E}_{v_{1:t-1}}\left(f\left(w^{(t)}\right)-f\left(w^{*}\right)\right)\\ =&\mathrm{E}_{v_{1:T}}\left(f\left(w^{(t)}\right)-f\left(w^{*}\right)\right)\end{aligned}\tag{3.4}$$

Hence combining (3.4), (3.3), and (3.2)

$$\mathrm{E}\left(f\left(w_{\mathrm{SGD}}^{(T)}\right)-f\left(w^{*}\right)\right)\leq\frac{1}{T}\sum_{t=1}^{T}\mathrm{E}\left(\langle w^{(t)}-w^{*},v_{t}\rangle\right)$$

Then Lemma 17 in Lect. notes 4 "Gradient descent" implies

$$\mathrm{E}\left(f\left(w_{\mathrm{SGD}}^{(T)}\right)-f\left(w^{*}\right)\right)\leq\mathrm{E}\left(\frac{1}{T}\frac{\left\|w^{*}\right\|^{2}}{2\eta}+\frac{\eta}{2}\frac{1}{T}\sum_{t=1}^{T}\left\|v_{t}\right\|^{2}\right)=\frac{\mathrm{E}\left\|w^{*}\right\|^{2}}{2\eta T}+\frac{\eta}{2}\frac{1}{T}\sum_{t=1}^{T}\mathrm{E}\left\|v_{t}\right\|^{2}$$

□

Note 16. The upper bound in (3.1) depends on $\mathrm{E}\left\|v_{t}\right\|^{2}$ which has to be bounded and hence the variation of v_t because

$$\mathrm{E}\left\|v_{t}\right\|^{2}=\mathrm{E}\left(\left\|v_{t}-\mathrm{E}\left(v_{t}\right)\right\|^{2}\right)+\left\|\mathrm{E}\left(v_{t}\right)\right\|^{2}=\mathrm{tr}\left(\mathrm{Var}\left(v_{t}\right)\right)+\left\|\mathrm{E}\left(v_{t}\right)\right\|^{2}\tag{3.5}$$

where $\mathrm{E}\left(\left\|v_{t}-\mathrm{E}\left(v_{t}\right)\right\|^{2}\right)=\mathrm{tr}\left(\mathrm{Var}\left(v_{t}\right)\right)$ is the sum of all the variances at each dimension of v_t and $\left\|\mathrm{E}\left(v_{t}\right)\right\|^{2}$ is a finite constant as $\mathrm{E}\left(v_{t}\right)\in\partial_{w}f\left(w_{t}\right)$ is the unbiased estimator of the sub-gradient by construction.

Proposition 17. *(Cont. Proposition 15) Let $f(\cdot)$ be a convex function over the set $\mathcal{W}=\{w:\|w\|\leq B\}$. Let $E\|v_t\|^2\leq\rho^2$. Assume we run SGD algorithm of $f(\cdot)$ with learning rate $\eta_t=\sqrt{\frac{B^2}{\rho^2 T}}$ for T steps, and output $w_{\mathrm{SGD}}^{(T)}=\frac{1}{T}\sum_{t=1}^{T}w^{(t)}$. Then*

(1) *upper bound on the sub-optimality is*

$$\mathrm{E}\left(f\left(w_{\mathrm{SGD}}^{(T)}\right)\right)-f\left(w^{*}\right)\leq\frac{B\rho}{\sqrt{T}}\tag{3.6}$$

(2) *a given level of accuracy ε such that $E\left(f\left(w_{SGD}^{(T)}\right)\right)-f\left(w^{*}\right)\leq\varepsilon$ can be achieved after T iterations*

$$T\geq\frac{B^{2}\rho^{2}}{\varepsilon^{2}}.$$

Proof. It follows from Proposition 15. Condition $\mathrm{E}\left\|v_{t}\right\|^{2}\leq\rho^{2}$ can be achieved, for instance by assuming Lipschitz (See Lemma 21 from "Lect. notes 4: Gradient descent"). □

4. Implementation of SGD in the learning Problem 1

Note 18. Note 19 implements SGD to the learning problem (Problem 1)

$$w^{*}=\arg\min_{w}\left(R_{g}\left(w\right)\right)=\arg\min_{w}\left(\mathrm{E}_{z\sim g}\left(\ell\left(h_{w},z\right)\right)\right)$$

Note 19. For a randomly drawn example $z\sim g(\cdot)$, the sub-gradient v of $\ell\left(w,z\right)$ at point w is an unbiased estimator of the sub-gradient of the risk $R_{g}\left(w\right)$ at point w. I.e. if $v\in\partial_{w}\ell\left(w,z\right)$ where $z\sim g(\cdot)$ then $\mathrm{E}_{z\sim g}\left(v|w\right)\in\partial_{w}R_{g}\left(w\right)$.

Proof. Let v be a sub-gradient of $\ell\left(w,z\right)$ at point w, then

$$\ell\left(u,z\right)-\ell\left(w,z\right)\geq\left\langle u-w,v\right\rangle \tag{4.1}$$

It is

$$\begin{aligned}R_{g}\left(u\right)-R_{g}\left(w\right)&=\mathrm{E}_{z\sim g}\left(\ell\left(u,z\right)-\ell\left(w,z\right)|w\right)\geq\mathrm{E}_{z\sim g}\left(\left\langle u-w,v\right\rangle|w\right)\\&=\left\langle u-w,\mathrm{E}_{z\sim g}\left(v|w\right)\right\rangle\end{aligned}$$

Hence, by definition, v is such that $\mathrm{E}_{z\sim g}\left(v|w\right)$ is a sub-gradient of $R_{g}\left(w\right)$. □

Note 20. Similarly, the average $\bar{v}=\frac{1}{m}\sum_{i=1}^{m}v_{i}$ of sub-gradients $v_{i}\in\partial_{w}\ell\left(w,\tilde{z}_{i}\right)$ at point w given a randomly drawn set of m examples $\left\{\tilde{z}_{i}\sim g(\cdot);\,i=1,...,m\right\}$ is an unbiased estimator of the risk function sub-gradient; i.e. $\mathrm{E}_{z\sim g}\left(\bar{v}|w\right)\in\partial_{w}R_{g}\left(w\right)$.

Example 21. Assume a differentiable loss function $\ell\left(\cdot,z\right)$ at each z and set sub-gradient $v=\nabla_{w}\ell\left(w,z\right)$, then

$$\mathrm{E}_{z\sim g}\left(v|w\right)=\mathrm{E}_{z\sim g}\left(\nabla_{w}\ell\left(h_{w},z\right)|w\right)=\nabla_{w}\mathrm{E}_{z\sim g}\left(\ell\left(h_{w},z\right)|w\right)=\nabla_{w}R_{g}\left(w\right)$$

Note 22. Assume there is available a finite dataset $\mathcal{S}_{n}=\left\{z_{i};i=1,...,n\right\}$ of size n which consists of independent realizations z_{i} from the data generating distribution g; $z_{i}\overset{\text{ind}}{\sim}g$. SGD (Algorithm 23) is an implementation of the SGD (Algorithm 6) in the learning Problem 1.

Algorithm 23. *Stochastic Gradient Descent with learning rate $\eta_{t}>0$, and batch size m, for Problem 1.*

For $t=1,2,3,...$ iterate:

1. *randomly generate a set $\mathcal{J}^{(t)}\subseteq\left\{1,...,n\right\}^{m}$ of m indices from 1 to n with or without replacement.*
2. *compute*

$$w^{(t+1)}=w^{(t)}-\eta_{t}v_{t}, \tag{4.2}$$

where $v_{t}=\frac{1}{\left|\mathcal{J}^{(t)}\right|}\sum_{j\in\mathcal{J}^{(t)}}\nu_{t,j}$ and $\nu_{t,j}\in\partial_{w}\ell\left(w^{(t)},z_{j}\right)$ for any $j\in\mathcal{J}^{(t)}$.

3. *terminate if a termination criterion is satisfied*

Note 24. If it is possible to sample anytime fresh examples z_i directly from the data generation model g instead of just having access to only a given finite dataset of examples $\mathcal{S}_n$, then step 1 in Algorithm 23 can become

(1) sample $\tilde{z}_j^{(t)} \sim g(\cdot)$ for $j = 1, ..., m$.

Note 25. Algorithm 23 may be called online SGD when using sub-samples of size one ($m = 1$) and batch SGD when using sub-samples of larger sizes ($m > 1$) .

Note 26. In theory, using larger batch size m has the benefit that reduces the variance of v_t at iteration t due to averaging effect, stabilizes the SGD algorithm, and reduces the error bound (3.1); see Remark 16.

Note 27. In practice, for a given fixed computational time, using smaller batch size m has the benefit that the algorithm iterates faster as each iteration processes less number of examples. E.g., consider the extreme cases GD vs online SGD utilized in a scenario of big-data (large training data set): if the dataset consists of several replications of the same values, GD (using all the data) has to process the same information multiple times, while the online SGD (using only one example at a time) would avoid this issue.

Note 28. In practice, for a given fixed computational time, it is possible for a SGD with smaller batch size m to present better generalization properties (wrt the theoretical assumptions) than those with larger m (GD is included). It is observed for the former to be often less prone to getting stuck in shallow local minima because of the additional amount of "noise" E.g., consider the extreme cases GD ($\mathcal{J}^{(t)} \equiv \{1, ..., n\}$) vs online SGD ($m = 1$) in a scenario with non-convex risk function (e.g. our theoretical assumptions as violated): if the Risk function presents local minima, considering less examples randomly chosen each time may cause fluctuations in the gradient that allow the chain to accidentally jump/escape to an area with a lower minimum.

Proposition 29. *(Implementing Proposition 17) Consider a convex-Lipschitz-bounded problem $(\mathcal{H}, \mathcal{Z}, \ell)$ with parameters ρ and B. Then for every $\epsilon > 0$, if we run SGD Algorithm 6 to minimize Problem 1 for $T \geq \frac{B^2}{\epsilon^2}\rho^2$ iterations then it produces output $w_{SGD}^{(T)}$ satisfying*

$$E\left(R_g\left(w_{SGD}^{(T)}\right)\right) \leq \min_{\forall w \in \mathcal{H}} \left(R_g\left(w\right)\right) + \epsilon$$

and hence we can achieve PAC guarantees.

Proof. It is just re-writing the formula in part 2 of Proposition 17. Notice that here, condition $\mathrm{E}\left\|v_t\right\|^2 \leq \rho^2$ is satisfied by the self-bounding property from Lipschitzness of the loss (see Lemma 21 from "Lect. notes 4: Gradient descent"); i.e. if ℓ is ρ-Lipschitz, then $\|v_t\| \leq \rho$ where $\nu_t \in \partial_w \ell\left(w^{(t)}, z_j\right)$ at $j \in \mathcal{J}^{(t)}$. □

Example 30. [1] We continue Example 26 in Section 4 of Lect. notes 4. Recall, we considered a hypothesis space $\mathcal{H}$ of linear functions $h : \mathbb{R}^2 \to \mathbb{R}$ with $h\left(w\right) = w_1 + w_2 x$, $w = \left(w_1, w_2\right)^\top$,

[1] https://github.com/georgios-stats/Machine_Learning_and_Neural_Networks_III_Epiphany_2025/tree/main/Lecture_notes/code/example_SGD.R

and $\ell\left(w, z=(x,y)^\top\right) = (y_i - w_1 - w_2 x)^2$. Here we consider a big dataset $\mathcal{S}_n = \{z_1, ..., z_n\}$ with $n = 10^6$ examples.

The batch SGD algorithm (Algorithm 23) with learning rate η_t and batch size $m = 10$ is
For $t = 1, 2, 3, ...$ iterate:

(1) Randomly generate a set $\mathcal{J}^{(t)}$ by drawing $m = 10$ numbers from $\left\{1, ..., n = 10^6\right\}$

(2) compute

$$w^{(t+1)} = w^{(t)} - \eta_t v_t,$$

where

$$v_t = \begin{pmatrix} 2w_1^{(t)} + 2w_2^{(t)} \frac{1}{m} \sum_{j\in\mathcal{J}^{(t)}} x_j - 2\frac{1}{m}\sum_{j\in\mathcal{J}^{(t)}} y_j \\ 2w_1^{(t)} \bar{x} + 2w_2^{(t)} \frac{1}{m} \sum_{j\in\mathcal{J}^{(t)}} x_j^2 - 2\sum_{j\in\mathcal{J}^{(t)}} y_j x_j \end{pmatrix}$$

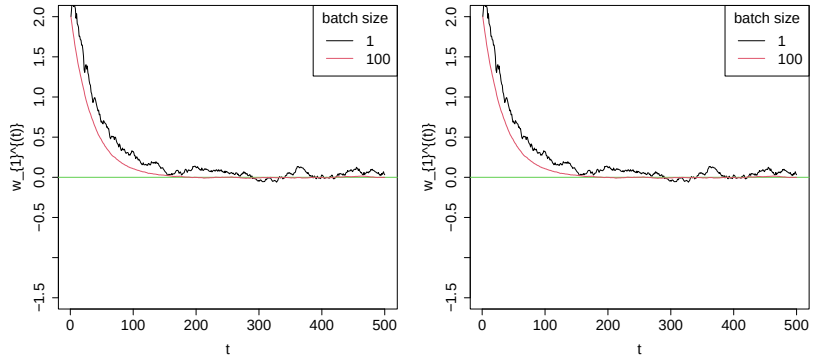
(3) if $t \geq T = 1000$ STOP

because

$$v_t = \frac{1}{m}\sum_{j\in\mathcal{J}^{(t)}} \nabla_w \ell\left(w^{(t)}, z_j\right) = ... = \begin{pmatrix} 2w_1^{(t)} + 2w_2^{(t)} \frac{1}{m} \sum_{j\in\mathcal{J}^{(t)}} x_j - 2\frac{1}{m}\sum_{j\in\mathcal{J}^{(t)}} y_j \\ 2w_1^{(t)} \bar{x} + 2w_2^{(t)} \frac{1}{m} \sum_{j\in\mathcal{J}^{(t)}} x_j^2 - 2\sum_{j\in\mathcal{J}^{(t)}} y_j x_j \end{pmatrix}$$

In Figures A.1a & A.1b, we observe that increasing the batch size has improved the convergence however this is not a panacea. Also it had reduced the oscillations of chain $\left\{w^{(t)}\right\}$.

Appendix A. Plots



(A) SGD for w_1 (B) SGD for w_2

Figure A.1. Simulations of the Example 30