普适性在线凸优化

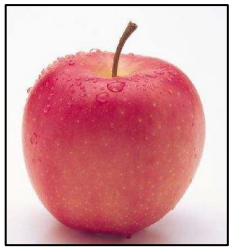
张利军
南京大学
人工智能学院

目录

- 研究背景
- 在线凸优化
- 普适性算法
- 总结与展望

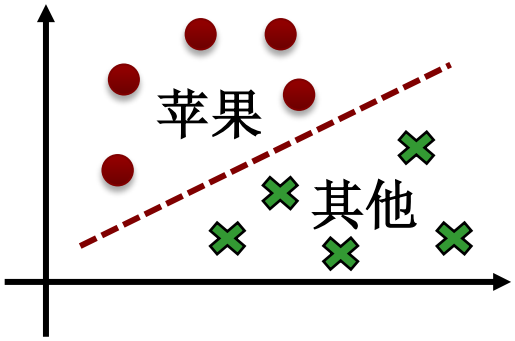
机器学习

- 监督学习



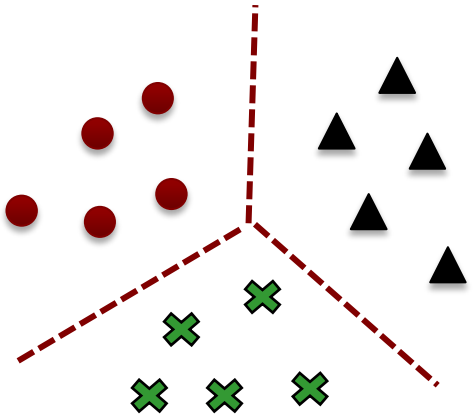
苹果

$$(\mathbf{x}_1, y_1)$$
$$\ldots$$
$$(\mathbf{x}_n, y_n) \Rightarrow y \approx h(\mathbf{x})$$



- 无监督学习

$$\mathbf{x}_1$$
$$\ldots$$
$$\mathbf{x}_n \Rightarrow h(\mathbf{x})$$



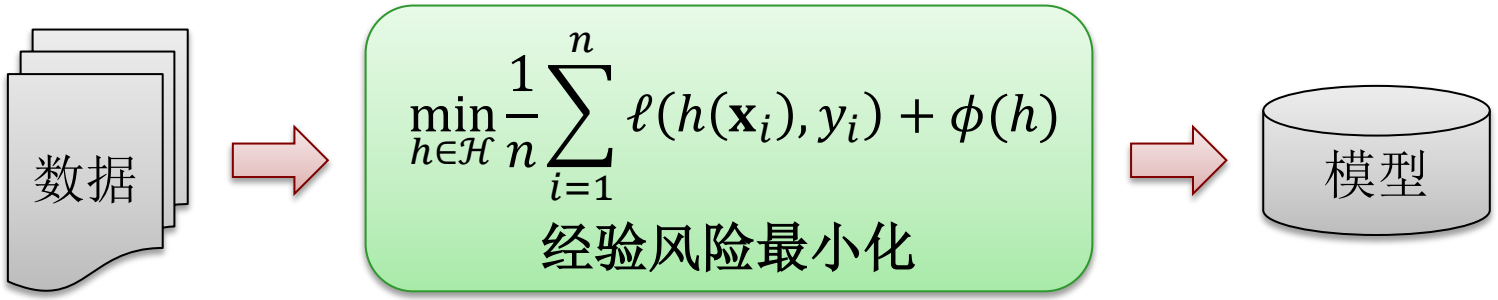
监督学习

- 经验风险最小化

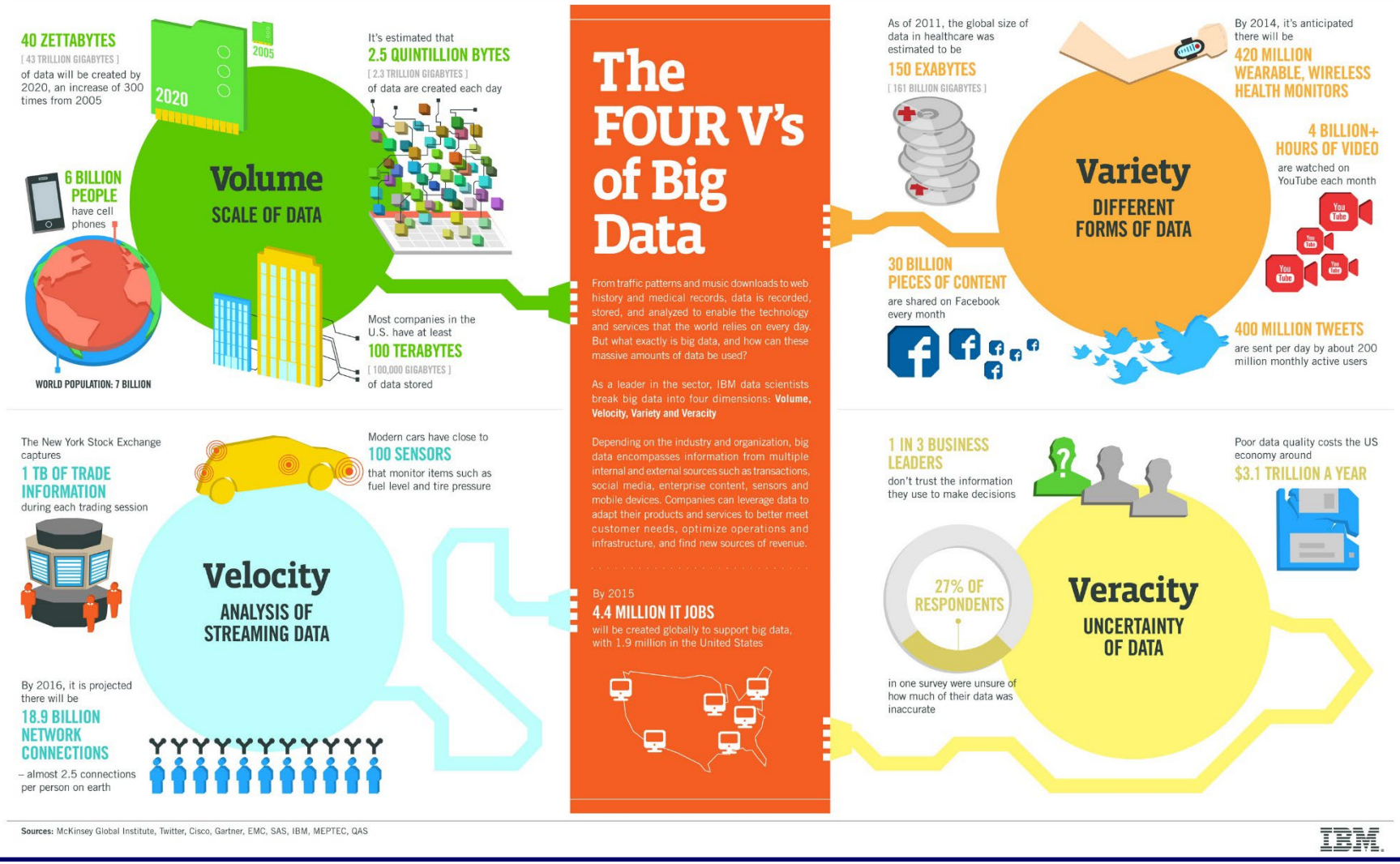
$$\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(h(\mathbf{x}_i), y_i) + \phi(h)$$

  - $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$是训练数据
  - n是样本数量、d是数据维度、 $\mathcal{H}$为假设空间
  - $\ell(\cdot,\cdot)$为损失函数、$\phi(\cdot)$为正则化因子

- 批量学习



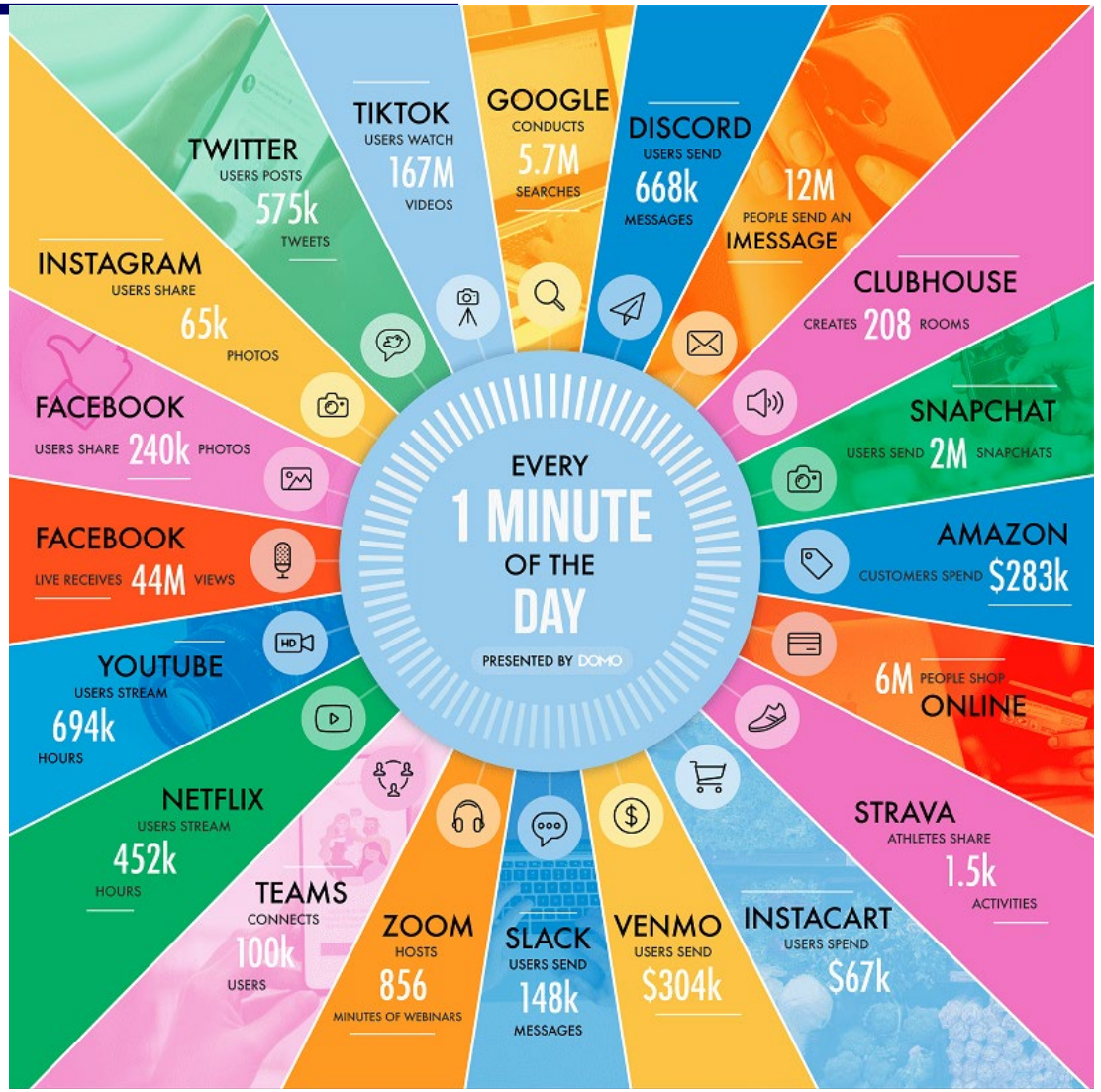
大数据



http://www.ibmbigdatahub.com/infographic/four-vs-big-data

快速增长的大数据

- Facebook
  - 24万张图片
- Google
  - 570万次搜索
- Amazon
  - 28.3万美金消费
- YouTube
  - 69.4万小时视频



https://www.socialmediatoday.com/news/what-happens-on-the-internet-every-minute-2021-version-infographic/607586/

大数据带来的挑战

- 经验风险最小化

$$\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(h(\mathbf{x}_i), y_i) + \phi(h)$$

  - n是训练样本的数量
  - 训练复杂度：$O(n)$、$O(n^2)$

- 大数据“规模大”
  - 训练过程计算量大
- 大数据“增长快”
  - 频繁需要重新训练

在线学习

在线学习

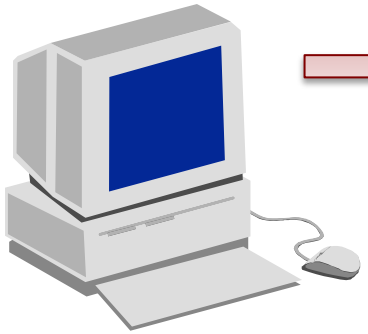
- 在线学习 [Cesa-Bianchi and Lugosi, 2006]

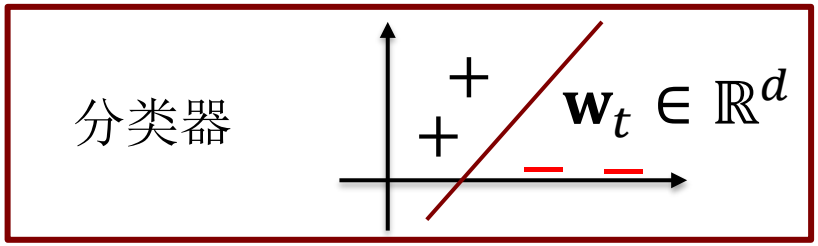
for $t = 1, 2, \ldots, T$ **do**

1. 学习器选择决策$\mathbf{w}_t \in \mathcal{W}$

   同时，对手选择函数$f_t(\cdot)$

end for





样本$(\mathbf{x}_t, y_t) \in \mathbb{R}^d \times \{\pm 1\}$

损失$f_t(\mathbf{w}) = \max(1 - y_t \mathbf{w}^\top \mathbf{x}_t, 0)$

在线学习

- 在线学习 [Cesa-Bianchi and Lugosi, 2006]

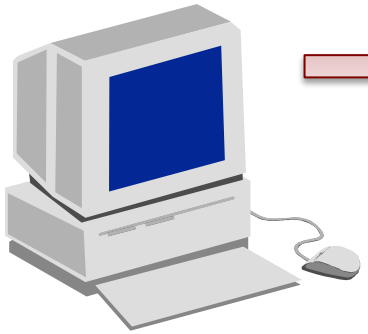
for $t = 1, 2, \ldots, T$ **do**

1. 学习器选择决策$\mathbf{w}_t \in \mathcal{W}$

   同时，对手选择函数$f_t(\cdot)$

2. 学习器遭受损失$f_t(\mathbf{w}_t)$，并更新$\mathbf{w}_t$

end for



遭受损失 $f_t(\mathbf{w}_t)$，更新 $\mathbf{w}_t$

在线学习

- 在线学习 [Cesa-Bianchi and Lugosi, 2006]

  for $t = 1, 2, \ldots, T$ **do**

  1. 学习器选择决策$\mathbf{w}_t \in \mathcal{W}$

     同时，对手选择函数$f_t(\cdot)$

  2. 学习器遭受损失$f_t(\mathbf{w}_t)$，并更新$\mathbf{w}_t$

  end for

- **累积损失**

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t)$$

  - 最小化累积损失

在线学习

- 在线学习 [Cesa-Bianchi and Lugosi, 2006]

  for $t = 1, 2, \ldots, T$ **do**

  1. 学习器选择决策$\mathbf{w}_t \in \mathcal{W}$

     同时，对手选择函数$f_t(\cdot)$

  2. 学习器遭受损失$f_t(\mathbf{w}_t)$，并更新$\mathbf{w}_t$

  end for

- **遗憾**（Regret）

$$\text{Regret} = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$$

批量学习

  - 最小化遗憾

在线学习

- 在线学习 [Cesa-Bianchi and Lugosi, 2006]

  for $t = 1, 2, \ldots, T$ **do**

  1. 学习器选择决策$\mathbf{w}_t \in \mathcal{W}$

     同时，对手选择函数$f_t(\cdot)$

  2. 学习器遭受损失$f_t(\mathbf{w}_t)$，并更新$\mathbf{w}_t$

  end for

- **遗憾**（Regret）

$$\text{Regret} = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) = o(T)$$

  - 最小化遗憾

目录

- 研究背景
- 在线凸优化
- 普适性算法
- 总结与展望

在线分类

➢ 感知机（Perceptron） [Rosenblatt, 1958]

for $t = 1, 2, \ldots, T$ **do**

1. 收到训练样本$(\mathbf{x}_t, y_t)$
2. 如果 $y_t\left(\mathbf{w}_t^{\mathsf{T}}\mathbf{x}_t\right) \geq 0$

$$\mathbf{w}_{t+1} = \mathbf{w}_t$$

3. 否则

$$\mathbf{w}_{t+1} = \mathbf{w}_t + y_t\mathbf{x}_t$$

end for

◘数据线性可分时算法收敛

在线凸优化

Classic Paper Prize of 2013

- 在线凸优化 [Zinkevich, ICML 2003]

  for $t = 1, 2, \ldots, T$ **do**

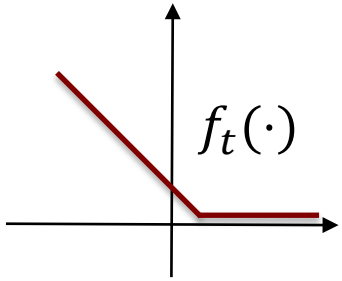
  1. 学习器选择决策$\mathbf{w}_t \in \mathcal{W}$

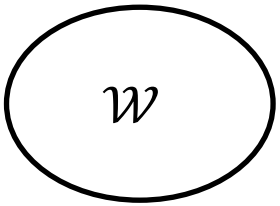
     同时，对手选择函数$f_t(\cdot)$

  2. 学习器遭受损失$f_t(\mathbf{w}_t)$，并更新$\mathbf{w}_t$

  end for

  - $f_t(\cdot)$是凸函数、$\mathcal{W}$是凸集合





- 丰富的理论和应用
  - 凸函数、强凸函数、指数凹函数
  - 基于凸损失的在线回归、在线分类
  - 随机优化（深度学习）

代表性算法

➢ 在线梯度下降 [Zinkevich, ICML 2003]

for $t = 1, 2, \ldots, T$ **do**

1. 看到损失函数$f_t(\cdot)$
2. 遭受损失$f_t(\mathbf{w}_t)$
3. 更新$\mathbf{w}_t$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_t)$$

end for

□$\nabla f_t(\mathbf{w}_t)$为函数$f_t(\cdot)$在$\mathbf{w}_t$处的梯度

□$\eta_t > 0$是步长

➢ 数据独立同分布时，即为随机梯度下降（SGD）

理论保障（1）

- 凸函数 [Zinkevich, ICML 2003]
  - 在线梯度下降，$\eta_t = O(1/\sqrt{t})$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) = O\left(\sqrt{T}\right)$$

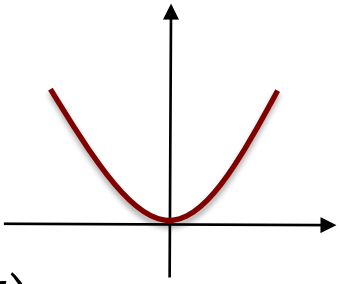
- 应用场景
  - 在线回归

$$f_t(\mathbf{w}) = (\mathbf{w}^\top \mathbf{x}_i - y_i)^2, \qquad \left|\mathbf{w}^\top \mathbf{x}_i - y_i\right|$$

  - 在线分类

$$f_t(\mathbf{w}) = \max(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i), \qquad \log(1 + \exp(-y_i \mathbf{w}^\top \mathbf{x}_i))$$

理论保障（2）



- 强凸函数 [Shalev-Shwartz et al., ICML 2007]
  - 在线梯度下降，$\eta_t = O(1/[\lambda t])$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) = O\left(\frac{\log T}{\lambda}\right)$$

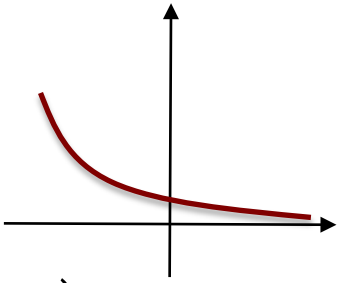
- 应用场景
  - 在线岭回归

$$f_t(\mathbf{w}) = (\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

  - 在线支持向量机

$$f_t(\mathbf{w}) = \max(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

理论保障（3）



- 指数凹函数 [Hazan et al., ML 2007]
  - 在线牛顿法（指数凹参数α已知）

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) = O\left(\frac{d \log T}{\alpha}\right)$$

- 应用场景
  - 在线最小二乘

$$f_t(\mathbf{w}) = (\mathbf{w}^\top \mathbf{x}_i - y_i)^2$$

  - 在线逻辑回归

$$f_t(\mathbf{w}) = \log(1 + \exp(-y_i \mathbf{w}^\top \mathbf{x}_i))$$

问题相关的理论保障（1）

- 小损失遗憾界 [Srebro et al., NIPS 2010]
  - 在线梯度下降、在线牛顿法

$$\text{Regret} = O\left(\sqrt{L_T^*}\right), \quad O\left(\frac{\log L_T^*}{\lambda}\right), \quad O\left(\frac{d \log L_T^*}{\alpha}\right)$$

  - $L_T^* = \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$，问题容易时遗憾

Best Student Paper

- 梯度变化遗憾界 [Chiang et al., COLT 2012]
  - 在线两次梯度下降 (OEGD)

$$\text{Regret} = O\left(\sqrt{V_T}\right), \quad O\left(\frac{\log V_T}{\lambda}\right), \quad O\left(\frac{d \log V_T}{\alpha}\right)$$

  - $V_T = \sum_{t=1}^{T} \max_{\mathbf{w}\in\mathcal{W}} \|\nabla f_t(\mathbf{w}) - \nabla f_{t-1}(\mathbf{w})\|^2$，函数变化缓慢

问题相关的理论保障（2）

➢ Adam [Kingma and Ba, ICLR 2015]

https://arxiv.org › cs

[1412.6980] Adam: A Method for Stochastic Optimization - arXiv

by DP Kingma · 2014 · Cited by 126165 — Abstract: We introduce **Adam**, an algorithm for first-order gradient-based **optimization** of **stochastic** objective functions, based on adaptive…

◘更新公式

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\nabla f_t(\mathbf{w}_t)$$

$$V_t = \beta_2 V_{t-1} + (1-\beta_2)\mathrm{diag}(\nabla f_t(\mathbf{w}_t)[\nabla f_t(\mathbf{w}_t)]^\top)$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\alpha}{\sqrt{t}} V_t^{-1/2}\, \mathbf{m}_t$$

◘理论保障：梯度相关的$O(\sqrt{T})$遗憾界

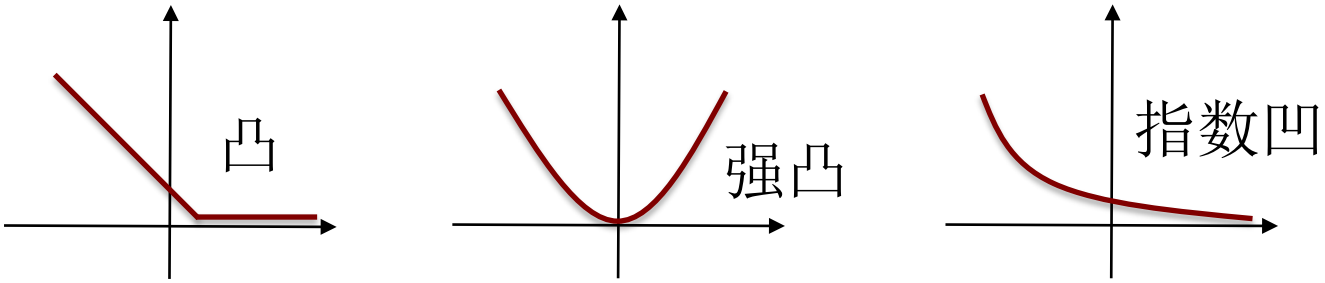
✓当梯度稀疏时自动变得更紧

目录

- 研究背景
- 在线凸优化
- 普适性算法
- 总结与展望

实际应用面临的挑战

- 现有在线学习算法
  - 凸函数：在线梯度下降，$\eta_t = O(1/\sqrt{t})$
  - 强凸函数：在线梯度下降，$\eta_t = O(1/[\lambda t])$
  - 指数凹函数：在线牛顿法，$\eta_t = O(1/[\alpha t])$



- 现有工作缺乏普适性
  - 针对不同类型的函数，需要选择不同的算法
  - 对于强凸、指数凹函数，还需要额外的参数

普适性算法

- Adaptive OGD [Hazan et al. NIPS 2008]
  - 支持凸函数、强凸函数
  1. 在线判断函数性质
  2. 在线调整OGD步长
  - 缺陷：在线计算强凸参数，不实用
- MetaGrad [van Erven and Koolen, NIPS 2016]
  - 支持凸函数、指数凹函数
  1. 构造替代损失，考虑$O(\log T)$个步长η

$$\ell_t^{\eta}(\mathbf{w}) = -\eta(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t) + \eta^2[(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t)]^2$$

  2. 运行$O(\log T)$个专家算法，通过元算法在线集成

我们的贡献（1）

- Maler [Wang et al., UAI 2019]
  - 凸函数替代损失

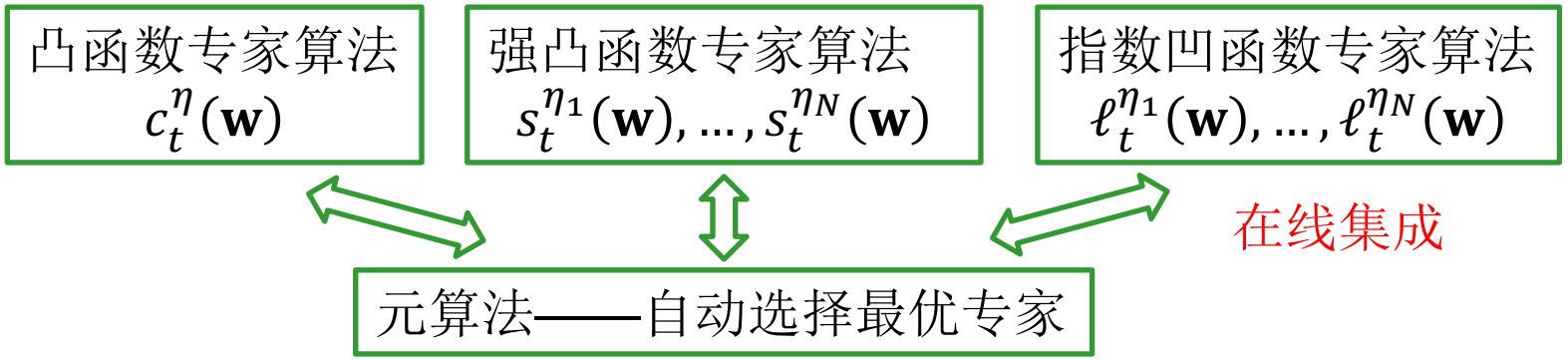
$$c_t^{\eta}(\mathbf{w}) = -\eta(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t) + (\eta GD)^2$$

  - 强凸函数替代损失

$$s_t^{\eta}(\mathbf{w}) = -\eta(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t) + \eta^2 G^2\|\mathbf{w}_t - \mathbf{w}\|^2$$

  - 指数凹函数替代损失

$$\ell_t^{\eta}(\mathbf{w}) = -\eta(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t) + \eta^2[(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t)]^2$$



我们的贡献（1）

- Maler [Wang et al., UAI 2019]
  - 凸函数替代损失

$$c_t^{\eta}(\mathbf{w}) = -\eta(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t) + (\eta GD)^2$$

  - 强凸函数替代损失

$$s_t^{\eta}(\mathbf{w}) = -\eta(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t) + \eta^2 G^2\|\mathbf{w}_t - \mathbf{w}\|^2$$

  - 指数凹函数替代损失

$$\ell_t^{\eta}(\mathbf{w}) = -\eta(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t) + \eta^2[(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t)]^2$$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) = \begin{cases} O(\sqrt{T}), & \text{凸函数} \\ O(\log T/\lambda), & \text{强凸函数} \\ O(d\log T/\alpha), & \text{指数凹函数} \end{cases}$$

我们的贡献（2）

小损失
遗憾界

LAMDA
Learning And Mining from DatA
http://lamda.nju.edu.cn

- UFO [Wang et al., AAAI 2020]
  - 凸且平滑函数替代损失

$$c_t^{\eta}(\mathbf{w}) = -\eta(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t) + (\eta D)^2\|\nabla f_t(\mathbf{w}_t)\|^2$$

  - 强凸且平滑函数替代损失

$$s_t^{\eta}(\mathbf{w}) = -\eta(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t) + \eta^2\|\nabla f_t(\mathbf{w}_t)\|^2\|\mathbf{w}_t - \mathbf{w}\|^2$$

  - 指数凹且平滑函数替代损失

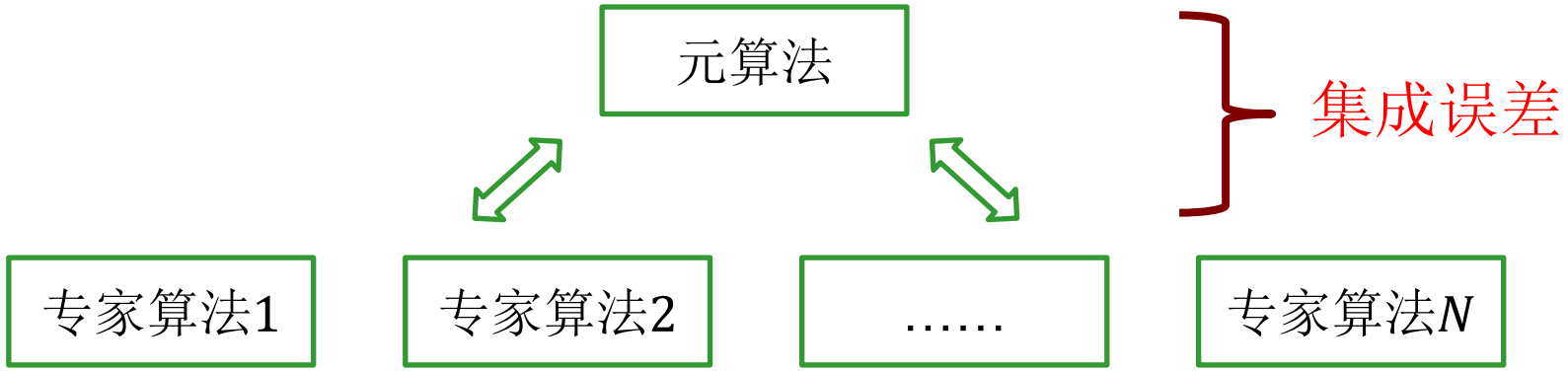
$$\ell_t^{\eta}(\mathbf{w}) = -\eta(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t) + \eta^2[(\mathbf{w}_t - \mathbf{w})^{\top}\nabla f_t(\mathbf{w}_t)]^2$$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) = \begin{cases} O\left(\sqrt{L_T^*}\right), & \text{凸函数} \\ O(\log L_T^*/\lambda), & \text{强凸函数} \\ O(d\log L_T^*/\alpha), & \text{指数凹函数} \end{cases}$$

我们的贡献（3）

- MetaGrad、Maler、UFO系列算法的局限
  - 依赖精心设计的替代函数，算法设计困难
  - 采用替代损失，专家算法无法利用原函数性质
  - 现有的各类在线学习算法无用武之地
  - 除了小损失界之外，难以得到其他的遗憾界

- 难点：集成引入额外误差，破坏专家算法最优性



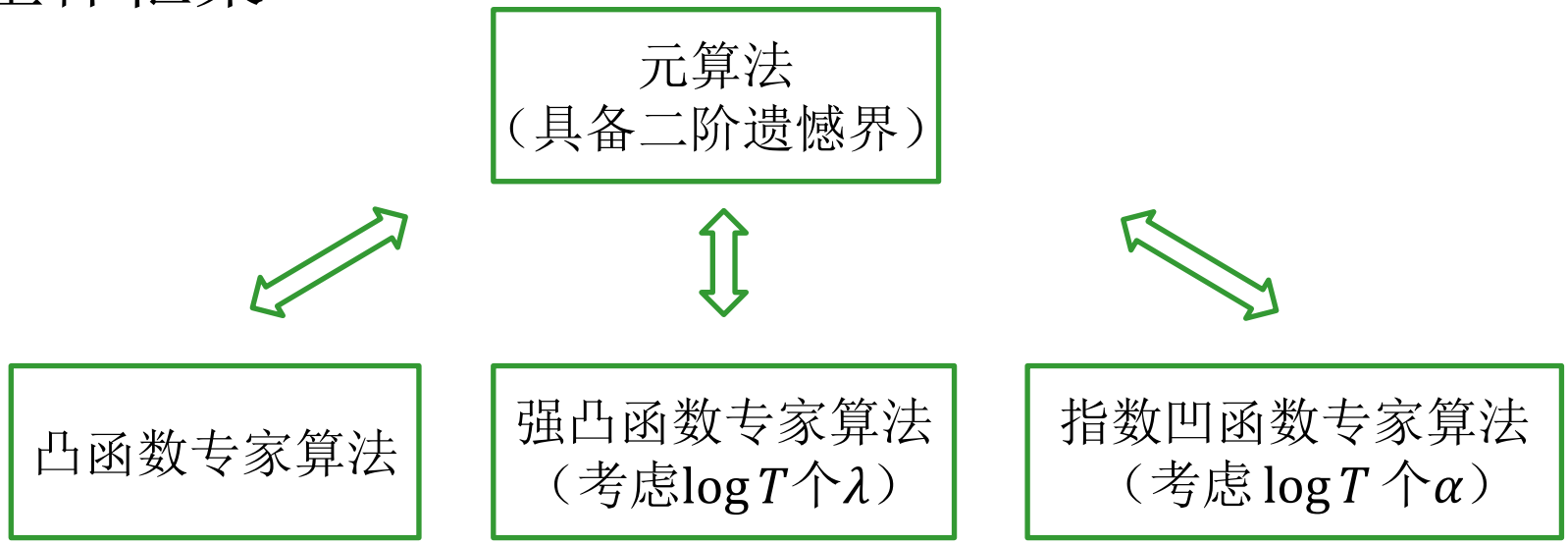
我们的贡献（3）

- MetaGrad、Maler、UFO系列算法的局限
  - 依赖精心设计的替代函数，算法设计困难
  - 采用替代损失，专家算法无法利用原函数性质
  - 现有的各类在线学习算法无用武之地
  - 除了小损失界之外，难以得到其他的遗憾界

- 普适性在线凸优化框架USC [Zhang et al., ICML 2022]
  - 避免引用替代损失，算法非常简单
  - 专家算法直接处理原函数，能够利用函数性质
  - 可以直接使用现有的各种在线学习算法
  - 对于强凸和指数凹函数，可以得到任何问题相关的遗憾界

普适性在线学习框架USC

➢ 整体框架



➢ 创新思想

1. 元算法采用线性损失，要求具备二阶遗憾界
2. 专家算法直接处理原函数，直接采用现有算法
3. 对于强凸和指数凹函数，考虑$\log T$个可能的λ和α

普适性在线学习框架USC

➢ 研究动机

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$$

$$= \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) + \sum_{t=1}^{T} f_t(\mathbf{u}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$$

元算法遗憾界
- 采用线性损失
- 具备二阶遗憾

专家算法遗憾界
- 直接利用专家算法的理论保障

元算法设计

- 线性损失函数

$$l_t(\mathbf{w}) = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle$$

  - 凸函数的一阶近似
  - 可用于控制元算法的遗憾，以强凸函数为例：

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq \sum_{t=1}^{T} l_t(\mathbf{w}_t) - \sum_{t=1}^{T} l_t(\mathbf{u}_t) - \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{u}_t\|^2$$

- 具备二阶遗憾界 → 能够利用强凸性 & 指数凹性

$$\sum_{t=1}^{T} l_t(\mathbf{w}_t) - \sum_{t=1}^{T} l_t(\mathbf{u}_t) = O\left(\sqrt{\sum_{t=1}^{T} \left(l_t(\mathbf{w}_t) - l_t(\mathbf{u}_t)\right)^2}\right)$$

抵消

元算法设计

- 线性损失函数

$$l_t(\mathbf{w}) = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle$$

  - 凸函数的一阶近似
  - 可用于控制元算法的遗憾，以强凸函数为例：

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{u}_t) \leq \sum_{t=1}^{T} l_t(\mathbf{w}_t) - \sum_{t=1}^{T} l_t(\mathbf{u}_t) - \frac{\lambda}{2} \|\mathbf{w}_t - \mathbf{u}_t\|^2$$

- 具备二阶遗憾界 → 能够利用强凸性&指数凹性
  - Adapt-ML-Prod [Gaillard et al., COLT 2014]
  - Squint+L [Mhammedi et al., COLT 2019]
  - MsMwC-Master [Chen et al., COLT 2021]

强凸函数专家算法

- 针对强凸函数的候选算法
  - 在线梯度下降 (SC-OGD) [Shalev-Shwartz et al., 2007]
  - ADAGRAD [Duchi et al., JMLR 2011]
  - 在线两次梯度下降 (OEGD) [Chiang et al., COLT 2012]
  - SAdam [Wang et al., ICLR 2020]
  - 针对小损失的S²OGD [Wang et al., AAAI 2020]
- 离散化强凸参数$\lambda \in [1/T, 1]$

$$\lambda \in \left\{\frac{1}{T}, \frac{2}{T}, \frac{2^2}{T}, \cdots, \frac{2^N}{T}\right\}, \qquad N = \lfloor \log_2 T \rfloor$$

- 为每个候选算法、每个离散λ构造一个专家
  - 计算复杂度仍然是$O(\log T)$

强凸函数理论保障

- **定理**：对于λ-强凸函数序列，并且$\lambda \in [1/T, 1]$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) = \min_{A} R\left(A, \hat{\lambda}\right) + O\left(\frac{\log\log T}{\lambda}\right)$$

  - 其中A表示候选强凸函数算法，$\hat{\lambda} \leq \lambda \leq 2\hat{\lambda}$

- 定理解读
  - 额外的$O\left(\frac{\log\log T}{\lambda}\right)$非常小，不影响最优性可以忽略
  - $\hat{\lambda} \leq \lambda \leq 2\hat{\lambda}$意味着算法可以自动估计未知的$\lambda$
  - $\min_{A} R\left(A, \hat{\lambda}\right)$意味着可以继承任何候选算法的遗憾界

强凸函数理论保障

- **定理**：对于λ-强凸函数序列，并且$\lambda \in [1/T, 1]$

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) = \min_{A} R\left(A, \hat{\lambda}\right) + O\left(\frac{\log\log T}{\lambda}\right)$$

  - 其中A表示候选强凸函数算法，$\hat{\lambda} \leq \lambda \leq 2\hat{\lambda}$

- 推论：当上述函数序列同时满足平滑且非负时

$$\text{Regret} = O\left(\frac{1}{\lambda}\left(\min(\log L_T^*, \log V_T) + \log\log T\right)\right)$$

  - 在小损失遗憾界和梯度变化遗憾界之中取最优

凸函数专家算法

- 针对凸函数的候选算法
  - 在线梯度下降 (OGD) [Zinkevich, ICML 2003]
  - ADAGRAD [Duchi et al., JMLR 2011]
  - 在线两次梯度下降 (OEGD) [Chiang et al., COLT 2012]
  - RMSprop [Tieleman and Hinton, 2012]
  - ADADELTA [Zeiler, 2012]
  - Adam [Kingma and Ba, ICLR 2015]
  - AO-FTRL [Mohri and Yang, AISTATS 2016]
  - SOGD [Zhang et al., ICML 2019]
- 为每个候选算法构造一个专家
  - 计算复杂度是$O(1)$

凸函数理论保障

- **定理**：对于一般的凸函数序列

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) = \min_{A} R(A) + \text{元算法遗憾}$$

$$= \min_{A} R(A) + O\left(\sqrt{T \log\log T}\right)$$

  - 其中A表示候选凸函数算法

- 定理解读
  - 元算法遗憾无法忽略，其在最坏情况下是$\sqrt{T \log\log T}$
  - 最终的遗憾界是子算法遗憾和元算法遗憾之和

凸函数理论保障

- **定理**：对于一般的凸函数序列

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) = \min_{A} R(A) + \text{元算法遗憾}$$

$$= \min_{A} R(A) + O\left(\sqrt{T \log\log T}\right)$$

  - 其中A表示候选凸函数算法

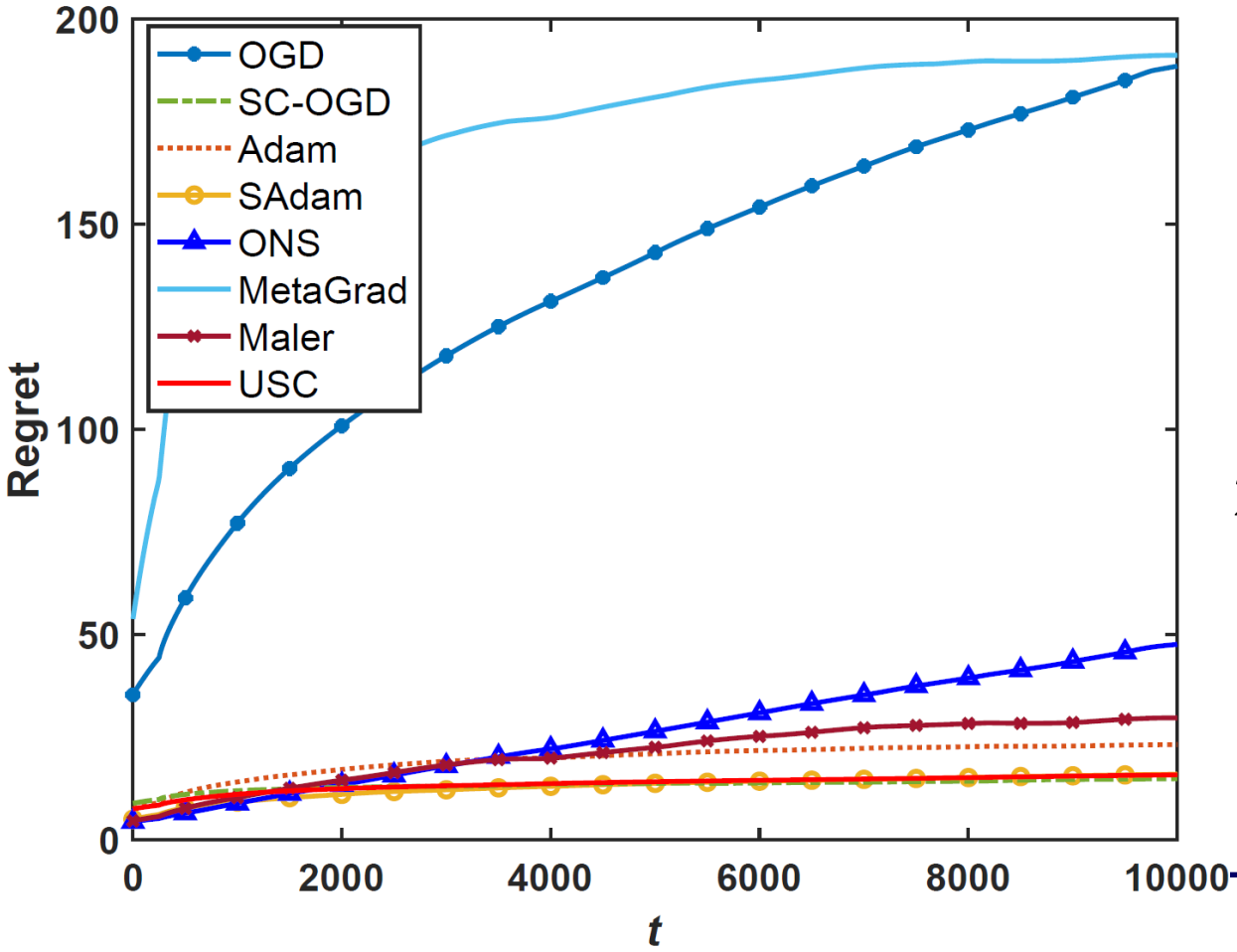
- 推论：当上述函数序列同时满足平滑且非负时

$$\text{Regret} = O\left(\sqrt{L_T^* \log\log T}\right)$$

  - 依然具备小损失遗憾界

实验结果——强凸函数

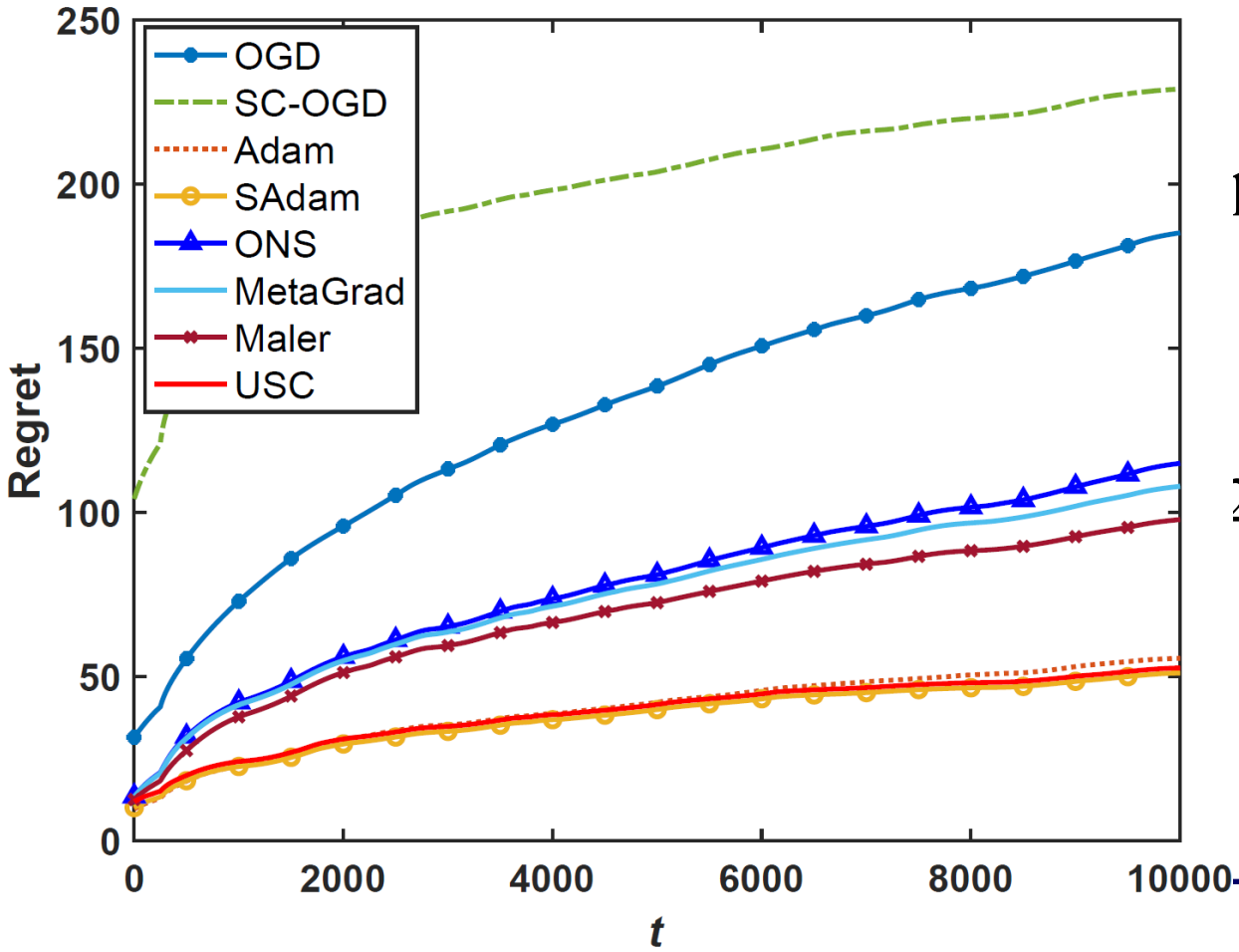
➢ 在线分类 $f_t(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} \max(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i) + \frac{\lambda}{2}\|\mathbf{w}\|^2$



1. USC性能接近最优的专家SC-OGD和SAdam
2. USC性能优于MetaGrad和Maler

实验结果——凸函数

- 在线分类 $f_t(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} \max(0, 1 - y_i \mathbf{w}^\top \mathbf{x}_i)$



1. USC性能接近最优的专家SAdam和Adam
2. USC性能优于MetaGrad和Maler

目录

- 研究背景
- 在线凸优化
- 普适性算法
- 总结与展望

总结与展望

- 普适性在线凸优化框架USC [Zhang et al., ICML 2022]
  - 专家算法处理原函数，直接使用现有在线算法
  - 元算法采用线性函数，要求具备二阶遗憾界
- USC的优势
  - 避免了设计替代损失，极其简单
  - 对于强凸和指数凹函数，得到问题相关的遗憾界
  - 对于凸函数，可以取得小损失遗憾界
- 未来工作
  - 拓展到无上界的函数或定义域
  - 拓展到动态环境：动态遗憾和自适应遗憾
  - 避免提前固定迭代轮数T

参考文献（1）

谢谢大家！

- Rosenblatt, F. (1958)
  The perceptron: a probabilistic model for information storage and organization in the brain.
  Psychological Review, 65:386–407.
- Zinkevich, M. (2003)
  Online convex programming and generalized infinitesimal gradient ascent.
  *In Proceedings of the 20th International Conference on Machine Learning (**ICML**).*
- Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2007)
  Pegasos: primal estimated sub-gradient solver for SVM.
  *In Proceedings of the 24th International Conference on Machine Learning (**ICML**).*
- Hazan, E., Agarwal, A., and Kale, S. (2007)
  Logarithmic regret algorithms for online convex optimization.
  Machine Learning, 69:169–192.

参考文献（2）

- Shalev-Shwartz, S. (2011)
  Online learning and online convex optimization.
  Foundations and Trends in Machine Learning, 4(2):107–194.
- Srebro, N., Sridharan, K., and Tewari, A. (2010).
  Smoothness, low-noise and fast rates.
  *In Advances in Neural Information Processing Systems 23 (**NIPS**).*
- Chiang, C.-K., Yang, T., Lee, C.-J., Mahdavi, M., Lu, C.-J., Jin, R., and Zhu, S. (2012).
  Online optimization with gradual variations.
  *In Proceedings of the 25th Annual Conference on Learning Theory (**COLT**).*
- Duchi, J., Hazan, E., and Singer, Y. (2011)
  Adaptive subgradient methods for online learning and stochastic optimization.
  Journal of Machine Learning Research, 12:2121–2159, 2011.
- Kingma, D. P. and Ba, J. (2015)
  Adam: A method for stochastic optimization.
  *In Proceedings of 3th International Conference on Learning Representations (**ICLR**).*

参考文献（3）

- Wang, G., Lu, S., Cheng, Q., Tu, W.-W., and Zhang, L. (2020)
  SAdam: A variant of Adam for strongly convex functions.
  *In International Conference on Learning Representations (**ICLR**).*
- Hazan, E., Rakhlin, A., and Bartlett, P. L. (2008)
  Adaptive online gradient descent.
  *In Advances in Neural Information Processing Systems 21 (**NIPS**).*
- van Erven, T. and Koolen, W. M. (2016)
  Metagrad: Multiple learning rates in online learning.
  *In Advances in Neural Information Processing Systems 29 (**NIPS**).*
- Wang, G., Lu, S., and Zhang, L. (2019)
  Adaptivity and optimality: A universal algorithm for online convex optimization.
  *In Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence (**UAI**).*
- Wang, G., Lu, S., Hu, Y., and Zhang, L. (2020)
  Adapting to smoothness: A more universal algorithm for online convex optimization.
  *In Proceedings of the 34th AAAI Conference on Artificial Intelligence (**AAAI**).*

参考文献（4）

- Zhang, L., Wang, G., Yi, J., and Yang, T. (2022)
  A simple yet universal strategy for online convex optimization.
  *In Proceedings of the 39th International Conference on Machine Learning (**ICML**).*
- Gaillard, P., Stoltz, G., and van Erven, T. (2014)
  A second-order bound with excess losses.
  *In Proceedings of the 27th Conference on Learning Theory (**COLT**).*
- Mhammedi, Z., Koolen, W. M., and Van Erven, T. (2019)
  Lipschitz adaptivity with multiple learning rates in online learning.
  *In Proceedings of the 32nd Conference on Learning Theory (**COLT**).*
- Chen, L., Luo, H., and Wei, C.-Y. (2021)
  Impossible tuning made possible: A new expert algorithm and its applications.
  *In Proceedings of 34th Conference on Learning Theory (**COLT**).*
- Orabona, F., Cesa-Bianchi, N., and Gentile, C. (2012)
  Beyond logarithmic bounds in online learning.
  *In Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (**AISTATS**).*

参考文献（5）

- Zeiler, M. D. (2012)
  Adadelta: An adaptive learning rate method.
  ArXiv e-prints, arXiv:1212.5701.
- Tieleman, T. and Hinton, G. (2012)
  Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude.
  COURSERA: Neural networks for machine learning, pages 26–31.
- Mohri, M. and Yang, S. (2016)
  Accelerating online convex optimization via adaptive prediction.
  *In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, (**AISTATS**).*
- Zhang, L., Liu, T.-Y., and Zhou, Z.-H. (2019)
  Adaptive regret of convex and smooth functions.
  *In Proceedings of the 36th International Conference on Machine Learning, (**ICML**).*