Efficient Online Set-valued Classification with Bandit Feedback

Zhou Wang, Xingye Qiao

Department of Mathematics and Statistics

Binghamton University
State University of New York

A Limitation of Conformal Prediction

- **(Class-specific) Conformal prediction** [Vovk et al., 2005, Vovk, 2012] returns a prediction set $\widehat{\mathcal{C}}(\boldsymbol{X})$ for an observation $(\boldsymbol{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ with the coverage guarantee

$$\mathbb{P}[Y \in \widehat{\mathcal{C}}(\boldsymbol{X}) \mid Y = k] \geq 1 - \alpha, \ \forall \alpha \in [0, 1].$$

- Given score functions $s(\boldsymbol{X}, k)$ and quantiles/thresholds $\tau_k, k \in \mathcal{Y}$, we have

$$\widehat{\mathcal{C}}(\boldsymbol{X}) := \{k \in \mathcal{Y} : s(\boldsymbol{X}, k) \geq \tau_k\}.$$

A Limitation of Conformal Prediction

- **(Class-specific) Conformal prediction** [Vovk et al., 2005, Vovk, 2012] returns a prediction set $\widehat{\mathcal{C}}(\boldsymbol{X})$ for an observation $(\boldsymbol{X}, Y) \in \mathcal{X} \times \mathcal{Y}$ with the coverage guarantee

$$\mathbb{P}[Y \in \widehat{\mathcal{C}}(\boldsymbol{X}) \mid Y = k] \geq 1 - \alpha, \ \forall \ \alpha \in [0, 1].$$

- Given score functions $s(\boldsymbol{X}, k)$ and quantiles/thresholds $\tau_k, k \in \mathcal{Y}$, we have

$$\widehat{\mathcal{C}}(\boldsymbol{X}) := \{k \in \mathcal{Y} : s(\boldsymbol{X}, k) \geq \tau_k\}.$$

- Conformal prediction requires fully observed label information:
  1. Fit a machine learning model $\boldsymbol{f}$ on **labeled** training data to obtain score functions $s(\boldsymbol{X}, k)$.
  2. Estimate quantiles τ_k for the score functions using **labeled** calibration data.

Online Bandit Feedback Settings

- Full label information is **absent** in online learning settings with **bandit feedback**, e.g., video recommendation and personalized medicine.
- In multi-class classification, a learner has no direct access to the label Y_t of the given instance $\boldsymbol{X}_t$ when updating the model.
    - The learner pulls an arm A_t and only receives the feedback $\mathbb{1}\{A_t = Y_t\}$.
    - Strategy to pull an arm: policy π_t, e.g., a probability distribution on $\mathcal{Y}$.

Online Learning with Bandit Feedback

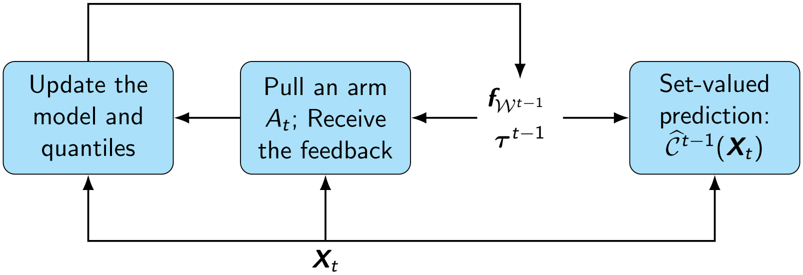

Figure: Flowchart of the online learning with bandit feedback. Here $\boldsymbol{\tau}^{t-1} = (\tau_1^{t-1}, \cdots, \tau_{|\mathcal{Y}|}^{t-1})^\top$.

Remark: Here $\boldsymbol{f}_{\mathcal{W}^{t-1}}$ is the based model (parameterized by $\mathcal{W}^{t-1}$) to construct score functions. $\tau_k^{t-1}, k \in \mathcal{Y}$ are estimated quantiles.

Estimate $\mathbb{1}\{Y_t = k\}$

- As a direct observation of Y_t is unavailable, we rely on an estimation to $\mathbb{1}\{Y_t = k\}$, i.e.,

$$\Delta_{t,k} := \frac{\mathbb{1}\{A_t = k\}}{\pi_t(k \mid \boldsymbol{X}_t)} \mathbb{1}\{A_t = Y_t\}.$$

Proposition 1

$\Delta_{t,k}$ serves as an unbiased estimator of $\mathbb{1}\{Y_t = k\}$. This is substantiated by the equation

$$\mathbb{E}_{\pi_t}\left[\Delta_{t,k}\right] = \mathbb{1}\{Y_t = k\},$$

where the expectation is taken with respect to policy π_t, conditioning on all previous information and the point $(\boldsymbol{X}_t, Y_t)$.

Train a Base Model

- Train a neural network $\boldsymbol{f}_{\mathcal{W}}(\boldsymbol{X}) = (f_{\mathcal{W}}^{1}(\boldsymbol{X}), \cdots, f_{\mathcal{W}}^{|\mathcal{Y}|}(\boldsymbol{X}))^\top \in \mathbb{R}^{|\mathcal{Y}|}$ with cross-entropy loss

$$\mathcal{L}(\boldsymbol{X}_t; \mathcal{W}) = -\sum_{k \in \mathcal{Y}} \mathbb{1}\{Y_t = k\} \cdot \log\left(\hat{p}(k \mid \boldsymbol{X}_t)\right),$$

where

$$\hat{p}(k \mid \boldsymbol{X}_t) := \frac{\exp(f_{\mathcal{W}}^{k}(\boldsymbol{X}_t))}{\sum_{\tilde{k} \in \mathcal{Y}} \exp(f_{\mathcal{W}}^{\tilde{k}}(\boldsymbol{X}_t))}, \ k \in \mathcal{Y}.$$

Train a Base Model

- Train a neural network $\boldsymbol{f}_{\mathcal{W}}(\boldsymbol{X}) = (f_{\mathcal{W}}^{1}(\boldsymbol{X}), \cdots, f_{\mathcal{W}}^{|\mathcal{Y}|}(\boldsymbol{X}))^{\top} \in \mathbb{R}^{|\mathcal{Y}|}$ with cross-entropy loss

$$\mathcal{L}(\boldsymbol{X}_t; \mathcal{W}) = -\sum_{k \in \mathcal{Y}} {\color{red}\Delta_{t,k}} \cdot \log\left(\hat{p}(k \mid \boldsymbol{X}_t)\right),$$

where

$$\hat{p}(k \mid \boldsymbol{X}_t) := \frac{\exp(f_{\mathcal{W}}^{k}(\boldsymbol{X}_t))}{\sum_{\tilde{k} \in \mathcal{Y}} \exp(f_{\mathcal{W}}^{\tilde{k}}(\boldsymbol{X}_t))}, \quad k \in \mathcal{Y}.$$

Train a Base Model

- Train a neural network $\boldsymbol{f}_{\mathcal{W}}(\boldsymbol{X}) = (f_{\mathcal{W}}^1(\boldsymbol{X}), \cdots, f_{\mathcal{W}}^{|\mathcal{Y}|}(\boldsymbol{X}))^\top \in \mathbb{R}^{|\mathcal{Y}|}$ with cross-entropy loss

$$\mathcal{L}(\boldsymbol{X}_t; \mathcal{W}) = -\sum_{k \in \mathcal{Y}} \Delta_{t,k} \cdot \log\left(\hat{p}(k \mid \boldsymbol{X}_t)\right),$$

where

$$\hat{p}(k \mid \boldsymbol{X}_t) := \frac{\exp(f_{\mathcal{W}}^k(\boldsymbol{X}_t))}{\sum_{\tilde{k} \in \mathcal{Y}} \exp(f_{\mathcal{W}}^{\tilde{k}}(\boldsymbol{X}_t))}, \quad k \in \mathcal{Y}.$$

and its updating rule

$$\mathcal{W}^t = \mathcal{W}^{t-1} - \eta_1 \nabla_{\mathcal{W}} \mathcal{L}(\boldsymbol{X}_t; \mathcal{W}^{t-1}).$$

Estimate Conformal Quantiles

- The check loss $\rho_\alpha(s,\tau) = (s-\tau)\cdot\big(\alpha - \mathbb{1}\{s<\tau\}\big)$ is used to find the $100\times\alpha\%$ quantile, τ, for the distribution of the score s. In particular, given the score function $s(\boldsymbol{X},k)$ for class $k\in\mathcal{Y}$, we aim to solve

$$\begin{aligned}\underset{\tau}{\operatorname{argmin}}\,\mathbb{E}\big[\rho_\alpha(s(\boldsymbol{X},k),\tau)\mid Y=k\big] &= \underset{\tau}{\operatorname{argmin}}\,\frac{\mathbb{E}\big[\mathbb{1}\{Y=k\}\cdot\rho_\alpha(s(\boldsymbol{X},k),\tau)\big]}{\mathbb{E}\big[\mathbb{1}\{Y=k\}\big]}\\ &= \underset{\tau}{\operatorname{argmin}}\,\mathbb{E}\big[\mathbb{1}\{Y=k\}\cdot\rho_\alpha(s(\boldsymbol{X},k),\tau)\big].\end{aligned}$$

- In practice, we instead work with its empirical counterpart

$$\mathbb{1}\{Y_t=k\}\cdot\rho_\alpha(s^{t-1}(\boldsymbol{X}_t,k),\tau).$$

Estimate Conformal Quantiles

- The check loss $\rho_\alpha(s,\tau) = (s-\tau)\cdot\big(\alpha - \mathbb{1}\{s<\tau\}\big)$ is used to find the $100\times\alpha\%$ quantile, τ, for the distribution of the score s. In particular, given the score function $s(\boldsymbol{X},k)$ for class $k\in\mathcal{Y}$, we aim to solve

$$\begin{aligned}\operatorname*{argmin}_{\tau}\mathbb{E}\big[\rho_\alpha(s(\boldsymbol{X},k),\tau)\mid Y=k\big] &= \operatorname*{argmin}_{\tau}\frac{\mathbb{E}\big[\mathbb{1}\{Y=k\}\cdot\rho_\alpha(s(\boldsymbol{X},k),\tau)\big]}{\mathbb{E}\big[\mathbb{1}\{Y=k\}\big]}\\ &= \operatorname*{argmin}_{\tau}\mathbb{E}\big[\mathbb{1}\{Y=k\}\cdot\rho_\alpha(s(\boldsymbol{X},k),\tau)\big].\end{aligned}$$

- In practice, we instead work with its empirical counterpart

$$\Delta_{t,k}\cdot\rho_\alpha(s^{t-1}(\boldsymbol{X}_t,k),\tau).$$

Estimate Conformal Quantiles

- The check loss $\rho_\alpha(s,\tau) = (s-\tau)\cdot\big(\alpha - \mathbb{1}\{s<\tau\}\big)$ is used to find the $100\times\alpha\%$ quantile, τ, for the distribution of the score s. In particular, given the score function $s(\boldsymbol{X},k)$ for class $k\in\mathcal{Y}$, we aim to solve

$$\begin{aligned}\operatorname*{argmin}_{\tau}\mathbb{E}\big[\rho_\alpha(s(\boldsymbol{X},k),\tau)\mid Y=k\big] &= \operatorname*{argmin}_{\tau}\frac{\mathbb{E}\big[\mathbb{1}\{Y=k\}\cdot\rho_\alpha(s(\boldsymbol{X},k),\tau)\big]}{\mathbb{E}\big[\mathbb{1}\{Y=k\}\big]}\\ &= \operatorname*{argmin}_{\tau}\mathbb{E}\big[\mathbb{1}\{Y=k\}\cdot\rho_\alpha(s(\boldsymbol{X},k),\tau)\big].\end{aligned}$$

- In practice, we instead work with its empirical counterpart

$$\Delta_{t,k}\cdot\rho_\alpha(s^{t-1}(\boldsymbol{X}_t,k),\tau).$$

and its updating rule

$$\tau_k^t = \tau_k^{t-1} + \eta_2\Delta_{t,k}\big(\alpha - \mathbb{1}\{s^{t-1}(\boldsymbol{X}_t,k) < \tau_k^{t-1}\}\big).$$

Algorithms

Algorithm 1: Bandit Conformal

Require: Initialize weight matrices $\mathcal{W}^0$, class-specific quantiles $\tau_k^0 = 0$, $k \in \mathcal{Y}$. A score function $s^t(\cdot,\cdot)$, a policy π_t and learning rates η_1, η_2.

1: **for** $t = 1, 2, 3, \cdots, T$ **do**

2: Learner receives a query $\boldsymbol{X}_t$

3: Generates a prediction set for the query: $\widehat{\mathcal{C}}^{t-1}(\boldsymbol{X}_t) := \left\{k \in \mathcal{Y} : s^{t-1}(\boldsymbol{X}_t, k) \geq \tau_k^{t-1}\right\}$

4: Learner pulls an arm $A_t \sim \pi_t$, receives the feedback $\mathbb{1}\{A_t = Y_t\}$, and computes $\Delta_{t,k}$

5: Update all weights and quantiles:

$$\begin{cases} \mathcal{W}^t = \mathcal{W}^{t-1} - \eta_1 \nabla_{\mathcal{W}} \mathcal{L}(\boldsymbol{X}_t; \mathcal{W}^{t-1}) \\ \tau_k^t = \tau_k^{t-1} + \eta_2 \Delta_{t,k} \left(\alpha - \mathbb{1}\{s^{t-1}(\boldsymbol{X}_t, k) < \tau_k^{t-1}\}\right) \end{cases}$$

6: **end for**

Remark: Choosing a proper η_2 might be challenging in practice [Gibbs and Candes, 2021].

Algorithms

Algorithm 2: Bandit Conformal with Experts

Require: Initialize weight matrices $\mathcal{W}^0$, class-specific quantiles $\tau_{j,k}^0 = 0$, and experts weights $\omega_{j,k}^0 = 1,\ j \in [J],\ k \in \mathcal{Y}$. A score function $s^t(\cdot,\cdot)$, a policy π_t and learning rates η_1, $\eta_{2,j}$.

for $t = 1, 2, 3, \cdots, T$ **do**

  Learner receives a query $\boldsymbol{X}_t$

  Generates a prediction set for the query: $\widehat{\mathcal{C}}^{t-1}(\boldsymbol{X}_t) := \left\{k \in \mathcal{Y} : s^{t-1}(\boldsymbol{X}_t, k) \geq \bar{\tau}_k^{t-1}\right\}$, where $\bar{\tau}_k^{t-1} = \sum_j \omega_{j,k}^{t-1} \tau_{j,k}^{t-1} / \sum_i \omega_{i,k}^{t-1}$

  Learner pulls an arm $A_t \sim \pi_t$, receives the feedback $\mathbb{1}\{A_t = Y_t\}$, and computes $\Delta_{t,k}$

  Update all weights and quantiles:

$$\begin{cases} \mathcal{W}^t = \mathcal{W}^{t-1} - \eta_1 \nabla_{\mathcal{W}} \mathcal{L}(\boldsymbol{X}_t; \mathcal{W}^{t-1}) \\ \tau_{j,k}^t = \tau_{j,k}^{t-1} + \eta_{2,j} \Delta_{t,k} \left(\alpha - \mathbb{1}\{s^{t-1}(\boldsymbol{X}_t, k) < \tau_{j,k}^{t-1}\}\right) \\ \omega_{j,k}^t = \exp\left(-\frac{1}{\sqrt{t+1}} \sum_{t' \leq t} \Delta_{t',k} \cdot \rho_\alpha\left(s^{t'-1}(\boldsymbol{X}_{t'}, k), \tau_{j,k}^{t'-1}\right)\right) \end{cases}$$

end for

Coverage Gap

Theorem 1

Define the filtration $\mathcal{F}_t := (\sigma(\boldsymbol{X}_t, Y_t) \times \sigma(\pi_t)) \cup \mathcal{F}_{t-1}$. Assume $\pi_t(k \mid \boldsymbol{X}_t) \geq c_k > 0$ for all $t \in [T]$ and $\mathbb{E}[\frac{\mathbb{1}\{Y_t = k\}}{\pi_t(k|\boldsymbol{X}_t)} \mid \mathcal{F}_{t-1}] = b_k^t$. With probability at least $1-\delta$ taken over all the randomness, for all class $k \in \mathcal{Y}$, Algorithm 1 yields the empirical coverage gap

$$CvgGap_k := \left| \alpha - \frac{1}{T_k} \sum_{t=1}^{T} \mathbb{1}\{Y_t = k\} \cdot \mathbb{1}\{Y_t \notin \widehat{\mathcal{C}}^{t-1}(\boldsymbol{X}_t)\} \right| \leq \frac{\tau_k^T}{\eta_2 T_k} + \frac{\zeta_k(T, \delta/|\mathcal{Y}|)}{T_k},$$

where $\zeta_k(T,\delta) = \frac{2}{3c_k} \log \frac{2}{\delta} + \sqrt{2 \log \frac{2}{\delta} \cdot \sum_{t=1}^{T} b_k^t}$, and $T_k = \sum_{t=1}^{T} \mathbb{1}\{Y_t = k\}$.

- The empirical coverage rate converges to the desired coverage rate α in the order of $\mathcal{O}(T^{-1/2})$ if $\eta_2 = \mathcal{O}(T^{-1/2})$ and $T_k = \mathcal{O}(T)$.

Regret Analysis for the Check Loss

Theorem 2

Let p_k be the prior probability of class $k \in \mathcal{Y}$, and $\tau_k^* = \operatorname{argmin}_\tau \frac{1}{T}\sum_{t=1}^T \mathbb{1}\{Y_t = k\}\rho_\alpha(s^{t-1}(\boldsymbol{X}_t), \tau)$ be the quantile estimate using all the data instances. Define the empirical regret associated with the check loss in the bandit feedback setting as $Reg_{k,\rho_\alpha}(T) := \frac{1}{T}\sum_{t=1}^T \Delta_{t,k}\rho_\alpha(s^{t-1}(\boldsymbol{X}_t), \tau_k^{t-1}) - \frac{1}{T}\sum_{t=1}^T \mathbb{1}\{Y_t = k\}\rho_\alpha(s^{t-1}(\boldsymbol{X}_t), \tau_k^*)$. By choosing $\eta_2 = \tau_k^* p_k^{1/2}\left(\sum_{t=1}^T \mathbb{E}\left[\frac{\mathbb{1}\{Y_t=k\}}{\pi_t^2(k|\boldsymbol{X}_t)}\right]\right)^{-1/2}$, Algorithm 1 yields an expected regret

$$\mathbb{E}[Reg_{k,\rho_\alpha}(T)] \le \frac{\tau_k^*}{T}\sqrt{p_k \sum_{t=1}^T \mathbb{E}\left[\frac{\mathbb{1}\{Y_t = k\}}{\pi_t^2(k \mid \boldsymbol{X}_t)}\right]}.$$

- The expected regret converges in the rate of $\mathcal{O}(T^{-1/2})$ if $\eta_2 = \mathcal{O}(T^{-1/2})$.

Experiments

- Set-up: BCCP is tested with three score functions (softmax, APS, RAPS) and two policies (softmax and uniform).
- Metrics: At each time t, metrics are computed on the accumulated batches $\mathcal{B}_s, s \leq t$. The coverage rate is set as 95%.
  - Accumulative Coverage Rate:

$$\text{Acum\_cvg\_min}(t) = \min_{k \in \mathcal{Y}} \text{Acum\_cvg}(t, k), \quad \text{Acum\_cvg\_max}(t) = \max_{k \in \mathcal{Y}} \text{Acum\_cvg}(t, k),$$

where

$$\text{Acum\_cvg}(t, k) = \frac{\sum_{s=1}^{t} \sum_{\boldsymbol{X}_i \in \mathcal{B}_s} \mathbb{1}\{Y_i = k \ \& \ Y_i \in \widehat{\mathcal{C}}^{t-1}(\boldsymbol{X}_i)\}}{\sum_{s=1}^{t} \sum_{\boldsymbol{X}_i \in \mathcal{B}_s} \mathbb{1}\{Y_i = k\}}$$

  - Accumulative Prediction Set Size:

$$\text{Acum\_size}(t) = \frac{\sum_{s=1}^{t} \sum_{\boldsymbol{X}_i \in \mathcal{B}_s} |\widehat{\mathcal{C}}^{t-1}(\boldsymbol{X}_i)|}{\sum_{s=1}^{t} |\mathcal{B}_s|}$$

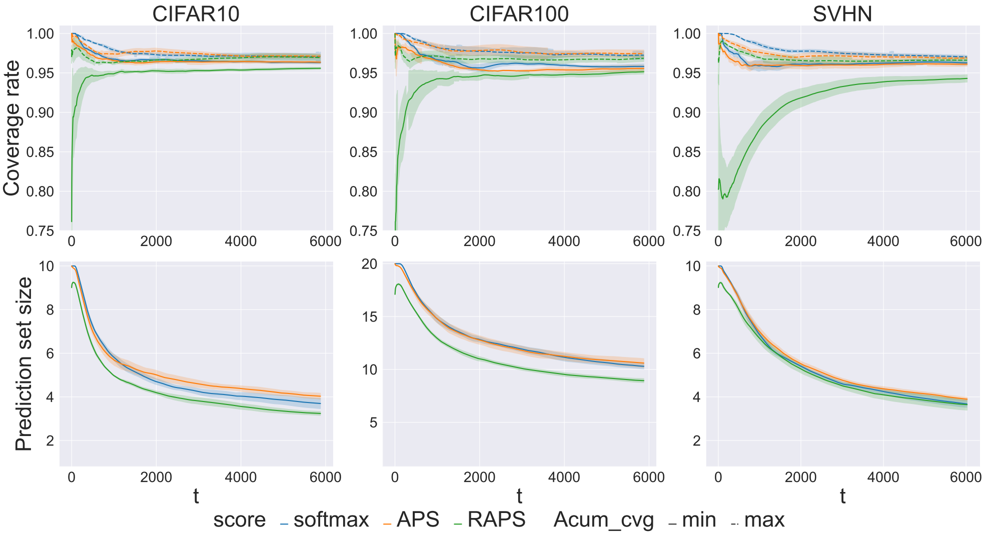

Figure: Performances under Algorithm 2 with softmax policy. The grid of learning rate is [0.1, 0.01, 0.001, 0.0001].

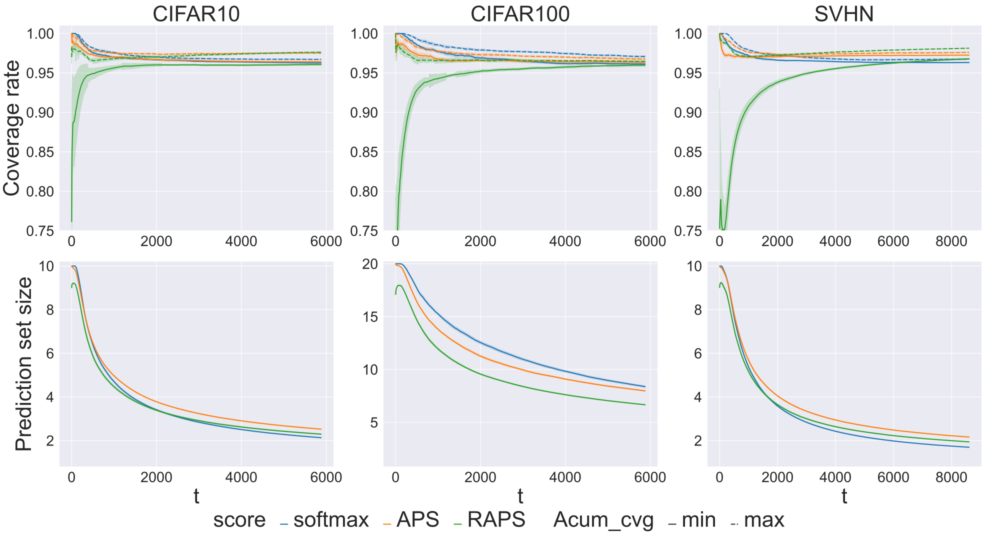

Figure: Performances under Algorithm 2 with uniform policy. The grid of learning rate is [0.1, 0.01, 0.001, 0.0001]

Conclusions

- The unbiased estimation with SGD allows the based model and thresholds to be efficiently updated in conformal prediction.
- The expert-based algorithm reduces the difficulty of selection of learning rate.
- Both coverage guarantee and the regret of the check loss converge at the rate of $\mathcal{O}(T^{-1/2})$.

References

Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490. PMLR, 2012.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.