Applied Quantitative Investment Management

Lecture 6: Entropy Pooling

Anton Vorobets

Agenda

- Section 5.1 and accompanying Python code
- Careful presentation of Entropy Pooling:
  - Its intuition, why it works, and some interesting features
  - How the problem is solved
  - Detailed description of common view specifications
- **Next lecture:** Sequential Entropy Pooling for better results
- **Lecture after that:** Causal and Predictive Market Views and Stress-Testing framework (Sequential Entropy Pooling combined with a Bayesian network)

Vague Entropy Pooling intuition

Theoretically sound view and stress-testing method similar to the Black-Litterman model but for fully general distributions, offering much more flexibility.

Real Entropy Pooling intuition

Theoretically sound method for updating fully general joint distributions in a predictive way, capable of handling a rich set of views and stress tests.

Entropy Pooling problem formulation

$$R = \begin{pmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,I} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,I} \\ \vdots & \vdots & \ddots & \vdots \\ R_{S,1} & R_{S,2} & \cdots & R_{S,I} \end{pmatrix} \quad p = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_S \end{pmatrix} \quad q = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_S \end{pmatrix}$$

$$q = \underset{x}{\operatorname{argmin}} \left\{ x^T \left(\ln x - \ln p \right) \right\}$$

$$\text{s.t.} \quad Gx \leq h \quad \text{and} \quad Ax = b$$

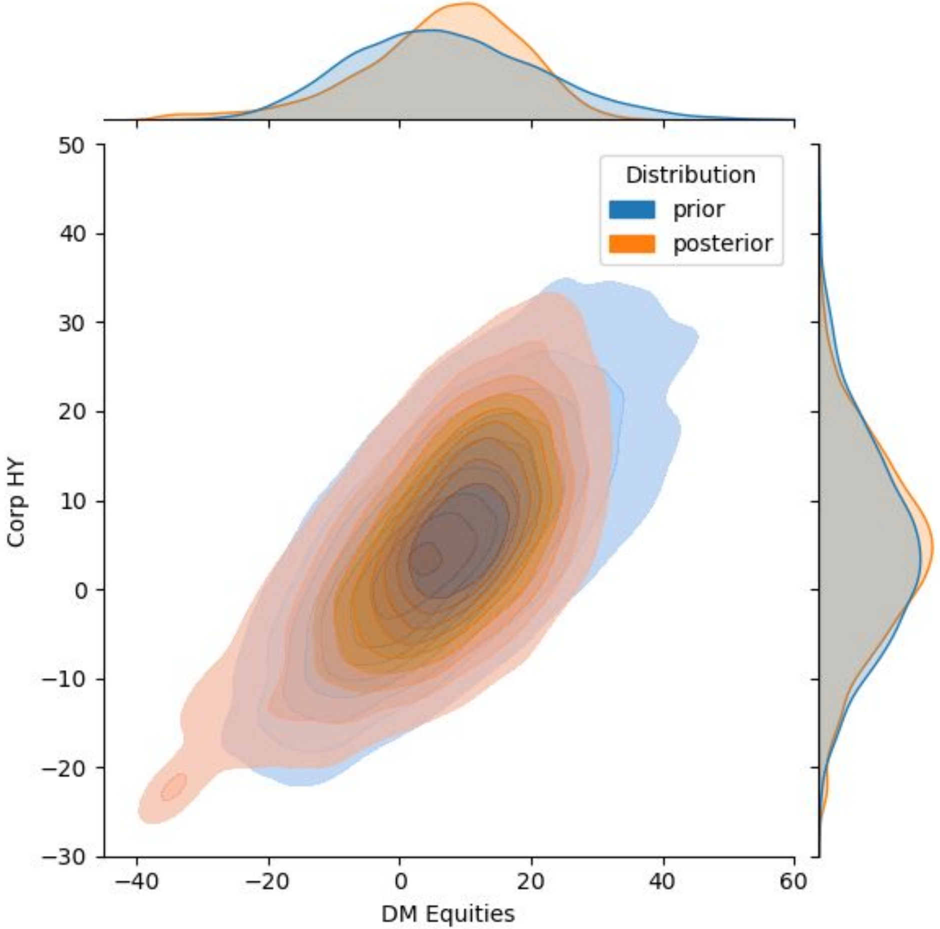
How an Entropy Pooling update looks

	DM Gov	Corp IG	Corp HY	EM Gov	DM Equities	EM Equities	Private Equity	Infrastructure	Real Estate	Hedge Funds	Put option	p	q
0	-2.901372	1.515755	7.243079	9.858614	27.987974	-7.091294	12.952589	0.607117	-2.051072	10.892261	-1.875000	0.01	0.009528
1	-4.033522	-5.507238	-17.472262	-18.084804	-20.901868	-42.943386	-24.630603	-4.877102	-6.808599	-8.255685	9.026868	0.01	0.010300
2	-2.112649	3.396251	3.433316	-2.171095	-0.207091	-24.640181	-4.598131	10.126871	-2.997219	2.751281	-1.875000	0.01	0.007035
3	4.953209	3.406907	27.820853	4.436701	25.658191	46.631981	75.017185	1.123291	10.170868	26.980165	-1.875000	0.01	0.002785
4	4.077621	-2.803989	1.067991	1.161560	19.527465	10.567691	27.660339	-4.110836	3.671376	10.251468	-1.875000	0.01	0.017757
5	5.028862	7.579274	14.766076	6.974784	-0.907956	-21.555335	5.031809	3.706813	-1.695980	0.211889	-1.875000	0.01	0.008335
6	-2.700348	-0.810961	-9.501812	-6.684766	1.070018	-6.555039	2.339627	6.119593	3.027697	5.498058	-1.875000	0.01	0.007617
7	3.976312	5.930553	4.328590	1.981325	13.535295	21.631855	-0.783067	11.487004	0.858899	10.237277	-1.875000	0.01	0.014521
8	-4.416313	-4.715980	-8.971976	-6.600808	-0.293693	-4.104135	17.010508	-14.610564	3.032070	3.442459	-1.875000	0.01	0.010635
9	2.115861	6.387309	14.575082	15.427097	11.223288	29.653759	29.527585	10.523831	5.674087	9.411179	-1.875000	0.01	0.018593
10	0.035257	1.025236	-5.285266	0.624084	-10.851014	-30.276292	-20.482211	-2.739756	0.922560	-6.129087	-1.023986	0.01	0.004728
11	1.974382	1.414184	16.825757	-2.814275	11.233995	19.782523	15.896183	-2.049651	0.240350	13.847232	-1.875000	0.01	0.012243
12	5.280698	0.871584	11.513783	3.309399	25.206057	18.720860	81.037608	9.508263	24.317649	19.498595	-1.875000	0.01	0.009221
13	2.257190	0.978109	4.755107	3.687936	1.943316	-2.218547	-11.355873	8.983795	6.558238	2.958365	-1.875000	0.01	0.013361

Why the relative entropy (KL divergence)?

Short answer: it has good properties for the updating problem.

Longer answer: it helps us update the prior distribution while predicting what will happen to all other assets and risk factors in a way that introduces the least amount of spuriosity.



Relation to maximum entropy

$$p = \begin{pmatrix} 1/S \\ 1/S \\ \vdots \\ 1/S \end{pmatrix}$$

$$\Downarrow$$

$$q = \operatorname*{argmin}_x x^T (\ln x - \ln p)$$

$$q = \operatorname*{argmin}_x \{x^T \ln x - x^T \ln \mathbf{1/S}\}$$

$$q = \operatorname*{argmax}_x \{-x^T \ln x\}$$

The effective number of scenarios (ENS)

- Exponential of the entropy of a joint scenario probability vector

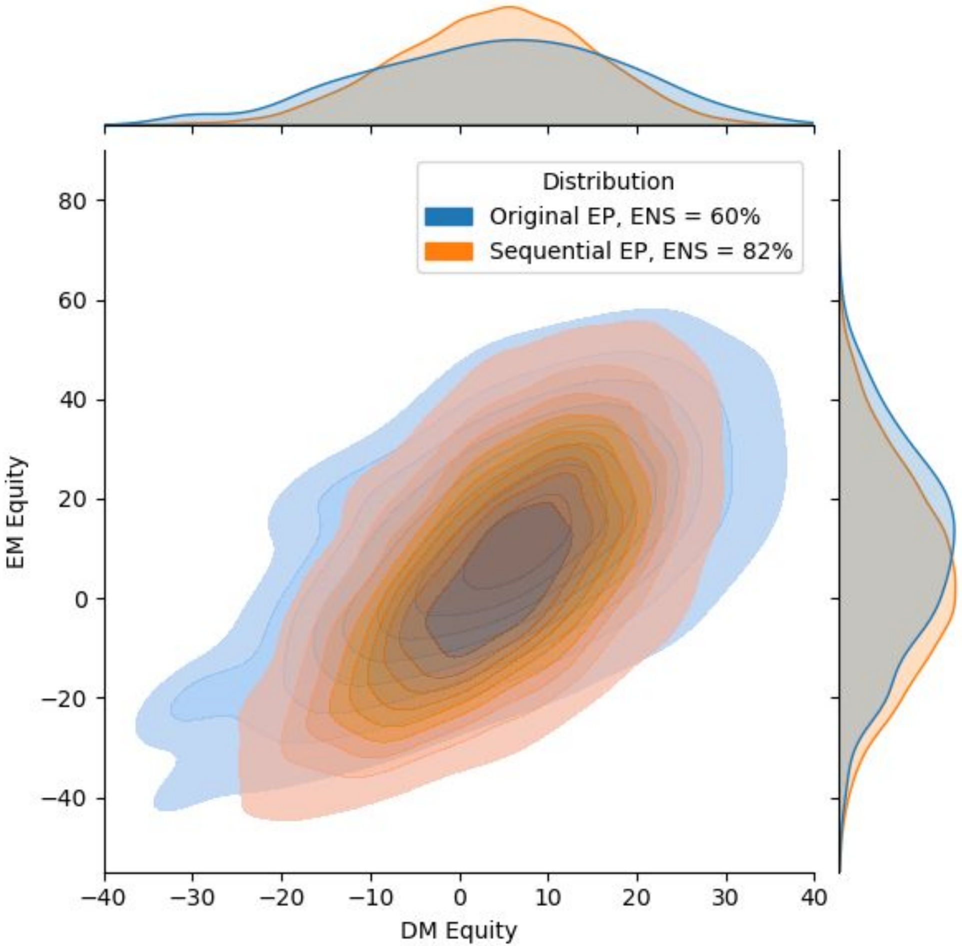
$$\hat{S} = \exp\left\{ -\sum_{s=1}^{S} q_s \ln q_s \right\}$$

- A measure of how concentrated the scenario probabilities are
- Equal to S when probabilities are uniform and 1 when all mass is assigned to one joint scenario
- In the typical uniform prior case, we have the relation that a lower effective number of scenarios leads to a higher relative entropy

ENS and relative entropy

When Entropy Pooling is applied in a clever sequential way (next lecture), we can get a lower relative entropy and therefore a higher effective number of scenarios for the same set of views.

This is equivalent to finding a lower minimum for some constrained minimization problem.



Solving the Entropy Pooling problem

- Primal Lagrangian formulation:

$$\mathcal{L}(x, \lambda, \nu) = x^T (\ln x - \ln p) + \lambda^T (Gx - h) + \nu^T (Ax - b)$$

- Solution from first order conditions:

$$x(\lambda, \nu) = \exp\left\{\ln p - \iota - G^T \lambda - A^T \nu\right\}$$

- **Issue:** potentially very high-dimensional because it has S variables, one for each joint market scenario
- **Solution:** solve the dual problem first

$$(\lambda^\star, \nu^\star) = \underset{\lambda \geq 0, \nu}{\operatorname{argmax}}\ \mathcal{G}(\lambda, \nu)$$

Entropy Pooling view specifications

- Views and stress tests implemented through constraints on posterior probabilities:

$$Gx \leq h \text{ and } Ax = b$$

- Fundamental principle:

$$f(R)^T x \gtreqqless c$$

- **Note:** views and stress tests can be nonlinear functions of the market simulation *R*, but must be formulated in a way so they are linear in the posterior probabilities
- Necessary for fast and stable solutions with high-dimensional simulations

Common view specifications

- Mean:

$$R_i^T x \gtreqqless \tilde{\mu}_i$$

- Variance:

$$R_i^T x = \tilde{\mu}_i \quad \text{and} \quad \left(R_i^T \odot R_i^T\right) x - \tilde{\mu}_i^2 \gtreqqless \tilde{\sigma}_i^2$$

- Skewness and kurtosis:

$$\left(\frac{R_i^T - \tilde{\mu}_i}{\tilde{\sigma}_i}\right)^3 x \gtreqqless \tilde{\gamma}_i \qquad \left(\frac{R_i^T - \tilde{\mu}_i}{\tilde{\sigma}_i}\right)^4 x \gtreqqless \tilde{\kappa}_i$$

- Correlation:

$$\frac{\left(R_i^T - \tilde{\mu}_i\right) \odot \left(R_j^T - \tilde{\mu}_j\right)}{\tilde{\sigma}_i \tilde{\sigma}_j} x \gtreqqless \tilde{\rho}_{ij}$$

Ranking views

- Mean ranking example:

$$R_i^T x - R_j^T x = (R_i - R_j)^T x \gtreqless 0$$

- Scaling mean ranking:

$$(R_i - aR_j)^T x \leq 0$$

- General formulation:

$$(f(R) - ag(R))^T x \leq 0$$

VaR and CVaR views

- VaR views:

$$a_s = \begin{cases} 0 & \text{if} \quad R_{s,i} > -\tilde{v} \\ 1 & \text{if} \quad R_{s,i} \leq -\tilde{v} \end{cases} \qquad a_{i,\alpha} = (a_1, a_2, \ldots, a_S)$$

$$a_{i,\alpha} x \gtreqqless 1 - \alpha$$

- CVaR views:

$$a_{i,\alpha} x = 1 - \alpha \quad \text{and} \quad \left(R_i^T \odot a_{i,\alpha}\right) x \gtreqqless -(1-\alpha)\, \tilde{cv}$$

- Note that CVaR requires us to fix the number of scenarios below the VaR value and therefore implicitly the VaR

Why the CVaR constraint works

- Writing out the sample CVaR:

$$\mathbb{E}\left[R_i | R_i \leq -\bar{v}\right] = \frac{\sum_{s \in \mathcal{CV}} R_{s,i} x_s}{\sum_{s \in \mathcal{CV}} x_s} = \frac{\left(R_i^T \odot a_{i,\alpha}\right) x}{a_{i,\alpha} x} \gtreqqless \frac{-\left(1-\alpha\right) \tilde{cv}}{\left(1-\alpha\right)} = -\tilde{cv}$$

- See the accompanying code for Section 5.1.3 for practical examples of how to implement VaR and CVaR views
- See the accompanying code to the Sequential Entropy Pooling Heuristics article for how to implement mean, standard deviation, skewness, kurtosis, and correlations views