spikiness_calc

February 19, 2025

[1]:
```
'''
"title": "spikiness_calc",
"author": "sunnyxxy"
'''

import numpy as np
import matplotlib.pyplot as plt

# both sorted in D values
def spikiness(D_sorted, w_sorted):
    weighted_mean = (np.sum(D_sorted**5 * w_sorted) / np.sum(w_sorted))**(1 / 5)
    weighted_variance = (np.sum((D_sorted**8 - weighted_mean**8)**2 * w_sorted)
 ↪/ np.sum(w_sorted))**(1 / 8)

    return np.sqrt(weighted_variance) / weighted_mean

spikiness(np.array([2, 6, 8]), np.array([3, 5, 4]))
```

[1]:
```
1.0725472562597993
```

[2]:
```
# Here acc is acc_v2
def spikiness_multiplier(acc, sp):
    sigmoid_spikiness = 0.94 + 0.12 / (1 + np.exp(-20*(sp-1)))
    return sigmoid_spikiness*(2*(acc/100)**20 - 1) + 2 - 2*(acc/100)**20

# Test the multiplier function
print("g(99.6, 1.1):", spikiness_multiplier(99.6, 1.1))
print("g(94, 1.1):",   spikiness_multiplier(94, 1.1))
print("g(99.6, 0.9):", spikiness_multiplier(99.6, 0.9))
print("g(94, 0.9):",   spikiness_multiplier(94, 0.9))
```

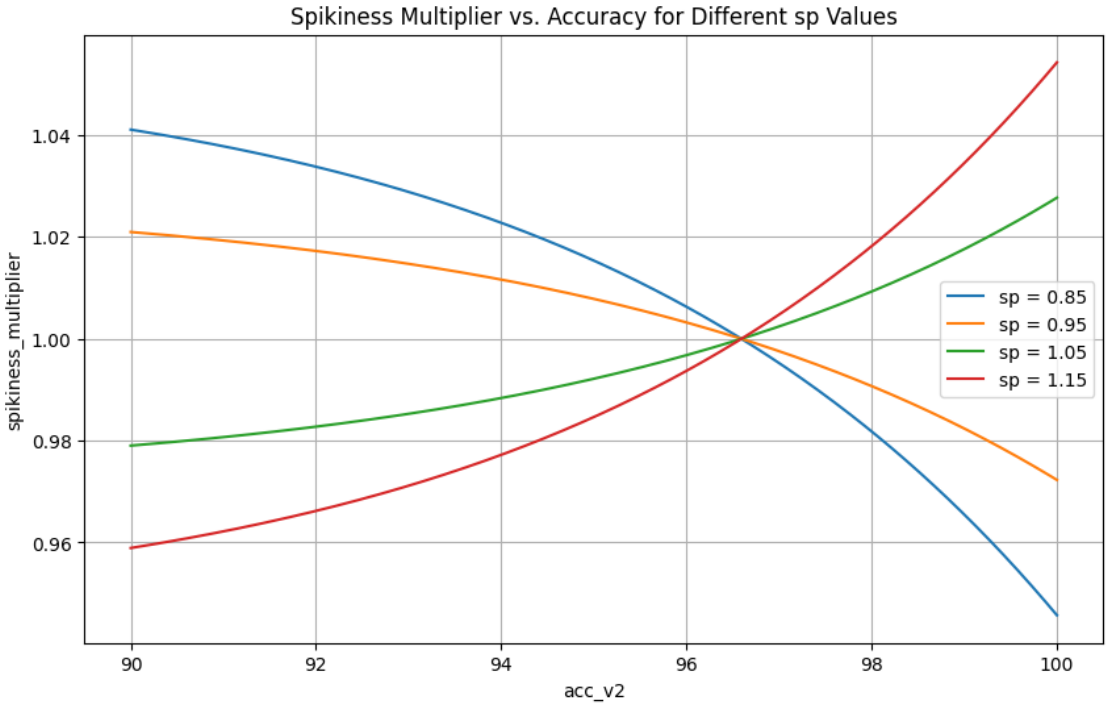
```
g(99.6, 1.1): 1.0386556190170215
g(94, 1.1): 0.980817536784896
g(99.6, 0.9): 0.9613443809829785
g(94, 0.9): 1.0191824632151043
```

[3]:
```
acc_values = np.linspace(90, 100, 400)
```

```python
# Define the list of v values for which we want to graph the function.
sp_values = [0.85, 0.95, 1.05, 1.15]

plt.figure(figsize=(10, 6))
for sp in sp_values:
    y = spikiness_multiplier(acc_values, sp)
    plt.plot(acc_values, y, label=f'sp = {sp}')

plt.xlabel('acc_v2')
plt.ylabel('spikiness_multiplier')
plt.title('Spikiness Multiplier vs. Accuracy for Different sp Values')
plt.legend()
plt.grid(True)
plt.show()
```



[ ]: